



ESCUELA TÉCNICA SUPERIOR DE INGENIERÍA INFORMATICA

TRABAJO FIN DE GRADO

INGERNIERÍA INFORMÁTICA – INGENIERÍA DEL SOFTWARE

REALIZADO POR:

JUAN JESÚS CAMPOS GARRIDO

DIRIGIDO POR:

ANOTNIO JESÚS CAÑETE MARTÍN

DEPARTAMENTO:

MATEMÁTICA APLICADA I

Sevilla, Mayo de 2024

Resumen

Este Trabajo de Fin de Grado se centra en el desarrollo de una aplicación web visual donde los usuarios jueguen a un juego donde aparezca un pedazo de una imagen y a partir de este pedazo adivinen de que foto se trata.

Antes de comenzar el proyecto se investigaron las aplicaciones web existentes con un concepto parecido al de esta aplicación y se ha analizado en que se diferencia mi aplicación con respecto a las ya existentes.

Una vez una versión preliminar de la aplicación fue diseñada y desarrollada, surgieron nuevas ideas como crear dos modos de juego multijugador o añadir un módulo de administración, finalmente todas estas ideas fueron desarrolladas e incluidas en la aplicación.

La funcionalidad de la aplicación ha sido evaluada exhaustivamente mediante distintos métodos de prueba y además se ha automatizado esta evaluación gracias al uso de GitHub Actions.

Además, en este documento se identifican áreas para posibles mejoras o expansiones de la aplicación en un futuro. Estas oportunidades ofrecen la oportunidad de seguir desarrollando y mejorando la aplicación para atraer más público.

En último lugar, una vez la aplicación fue desarrollada al completo, también fue desplegada y se le dio acceso a ella a una serie de usuarios pilotos que la han probado y han dado sus opiniones junto a algunas propuestas de mejora.

Como colofón, se han realizado un pequeño estudio de las partidas que han jugado los usuarios piloto para evaluar la dificultad del juego y ver cuando los usuarios tienen mayor probabilidad de adivinar la imagen.

En resumen, este Trabajo de Fin de Grado ha terminado con el desarrollo exitoso de la aplicación CUPES <http://cupes.es>, esta aplicación muestra la aplicación práctica de todos los conocimientos adquiridos a lo largo de mi formación en el grado universitario.

Agradecimientos

Me gustaría agradecer a todos los compañeros con los que he trabajado a lo largo de estos cuatro años de carrera, ya que de alguna forma u otra he aprendido de todos y lo conseguido en este proyecto en parte también es gracias ellos. Especialmente me gustaría agradecer a los compañeros que más me han acompañado en estos dos últimos años ya que ha sido cuando más he aprendido y donde mejor me he sentido trabajando en equipo, además de que todos ellos han probado la aplicación desarrollada en este proyecto y me han propuesto mejoras para la misma. Entre esos compañeros están:

- Alejandro Campano Galán
- Antonio Carretero Díaz
- David Cortabarria Romero
- Pablo Mera Gómez
- Alejandro Santiago Félix

También me gustaría agradecer a mi tutor Antonio Jesús Cañete Martín por su apoyo en este proceso donde con sus sugerencias hemos ido mejorando el resultado y finalmente me encuentro muy orgulloso de lo conseguido. Principalmente me gustaría agradecerle su comprensión a la hora de hacer reuniones ya que a lo largo de este año he tenido un horario bastante complicado y el siempre ha sido muy comprensivo y me ha facilitado mucho el realizar reuniones.

Finalmente me gustaría agradecer a mi familia y amigos por sus constante comprensión y apoyo durante todo este proceso donde ha tenido que renunciar a pasar más tiempo con ellos para poder centrarme en este proyecto.

Este proyecto no habría sido posible sin el apoyo, comprensión y propuestas de mejora de todas estas personas con las que estoy sinceramente agradecido.

Índice general

Índice general	III
Índice de cuadros	IV
Capítulo 1 Definición de objetivos	1
Capítulo 2 Análisis de antecedentes y aportación realizada	3
Capítulo 3 Análisis temporal y de costes de desarrollo	7
Capítulo 4 Análisis de requisitos, diseño e implementación	15
Capítulo 5 Manual de usuario	63
Capítulo 6 Pruebas	87
Capítulo 7 Comparación con otras alternativas	93
Capítulo 8 Problemas encontrados	95
Capítulo 9 Conclusiones y desarrollos futuros	99
Capítulo 10 Bibliografía	107

Índice de cuadros

Cuadro 3.1:Resumen de tiempo por tarea.....	13
Cuadro 3.2: Costes del proyecto	14
Cuadro 4.1: Historia de usuario jugar partidas.....	15
Cuadro 4.2: Historia de usuario consultar estadísticas	15
Cuadro 4.3: Historia de usuario registro de usuarios	15
Cuadro 4.4: Historia de usuario consultar el manual	16
Cuadro 4.5: Historia de usuario administración de imágenes	16
Cuadro 4.6: Historia de usuario administración de jugadores	16
Cuadro 4.7: Historia de usuario estudio de estadísticas	16
Cuadro 4.8: Requisito general juego solitario	17
Cuadro 4.9: Requisito general juego local.....	17
Cuadro 4.10: Requisito general juego en línea.....	17
Cuadro 4.11: Requisito general gestión de usuarios	17
Cuadro 4.12: Requisito general módulo de administración.....	18
Cuadro 4.13: Requisito general exportación de estadísticas.....	18
Cuadro 4.14: Actor usuario no registrado	24
Cuadro 4.15: Actor usuario registrado	24
Cuadro 4.16: Actor usuario no registrado 2	25
Cuadro 4.17: Actor administrador.....	25
Cuadro 4.18: Caso de uso jugar una partida en solitario	25
Cuadro 4.19: Caso de uso jugar una partida uno contra uno en local	26
Cuadro 4.20: Caso de uso crear una partida en línea	27
Cuadro 4.21: Caso de uso unirse a una partida en línea	27
Cuadro 4.22: Caso de uso jugar una partida en línea.....	28
Cuadro 4.23: Caso de uso registro de usuarios	28
Cuadro 4.24: Caso de uso inicio de sesión	29
Cuadro 4.25: Caso de uso consultar mi perfil	29
Cuadro 4.26: Caso de uso consultar estadísticas generales	30
Cuadro 4.27: Caso de uso consultar mis estadísticas	30
Cuadro 4.28: Caso de uso ver listado de imágenes	31
Cuadro 4.29: Caso de uso consultar datos de una imagen	31
Cuadro 4.30: Caso de uso consultar el manual	32
Cuadro 4.31: Caso de uso edición del perfil.....	32
Cuadro 4.32: Caso de uso cambio de contraseña	33
Cuadro 4.33: Caso de uso añadir una nueva imagen	33
Cuadro 4.34: Caso de uso deshabilitar una imagen	34
Cuadro 4.35: Caso de uso habilitar una imagen.....	34
Cuadro 4.36: Caso de uso ver listado de jugadores	35
Cuadro 4.37: Caso de uso consultar datos de un jugador.....	35
Cuadro 4.38: Caso de uso deshabilitar un jugador	36
Cuadro 4.39: Caso de uso habilitar un jugador	36
Cuadro 4.40: Caso de uso exportar datos de un jugador	37
Cuadro 4.41: Caso de uso exportar datos de una imagen	37
Cuadro 4.42: Caso de uso exportar datos de partidas en solitario	38
Cuadro 4.43: Requisito de información sobre los usuarios.....	38
Cuadro 4.44: Requisito de información sobre las partidas en solitario	39
Cuadro 4.45: Requisito de información sobre las partidas multijugador en línea	39

Cuadro 4.46: Requisito de información sobre las partidas multijugador locales.....	40
Cuadro 4.47: Requisito de información sobre las imágenes	40
Cuadro 4.48: Regla de negocio sobre el tipo de partida	40
Cuadro 4.49: Regla de negocio sobre las estadísticas	41
Cuadro 4.50: Regla de negocio sobre las partidas en solitario	41
Cuadro 4.51: Regla de negocio sobre las partidas multijugador locales.....	41
Cuadro 4.52: Regla de negocio sobre las partidas en línea.....	41
Cuadro 4.53: Regla de negocio sobre el abandono de partidas.....	42
Cuadro 4.54: Regla de negocio contra la pasividad en partidas en línea.....	42
Cuadro 4.55: Regla de negocio sobre las partidas no comenzadas	42
Cuadro 4.56: Requisito de seguridad sobre la protección de usuarios.....	42
Cuadro 4.57: Requisito de seguridad protección SQLInjection.....	43
Cuadro 4.58: Requisito de seguridad protección ScriptInjection	43
Cuadro 4.59: Requisito de seguridad protección CSRF	43
Cuadro 4.60: Matriz de trazabilidad de los requisitos de información	43
Cuadro 4.61: Matriz de trazabilidad sobre los casos de uso	44
Cuadro 4.62: Matriz de trazabilidad sobre las reglas de negocio	45
Cuadro 4.63: Matriz de trazabilidad sobre los requisitos no funcionales	45
Cuadro 9.1: Ratio de acierto por país.....	102

Índice de figuras

Figura 4.1: Diagrama de caso de uso jugar en solitario.....	18
Figura 4.2: Diagrama de caso de uso registrarse.....	19
Figura 4.3: Diagrama de caso de uso iniciar sesión.....	19
Figura 4.4: Diagrama de caso de uso jugar uno contra uno local.....	20
Figura 4.5: Diagrama de caso de uso jugar partida en línea	20
Figura 4.6: Diagrama de caso de uso editar usuario	21
Figura 4.7: Diagrama de caso de uso consultar el manual	21
Figura 4.8: Diagrama de caso de uso consultar estadísticas	22
Figura 4.9: Diagrama de caso de uso añadir nueva imagen	22
Figura 4.10: Diagrama de caso de uso habilitar/deshabilitar imagen.....	23
Figura 4.11: Diagrama de caso de uso habilitar/deshabilitar usuario.....	23
Figura 4.12: Diagrama de caso de uso exportar datos	24
Figura 4.13: Diagrama de la arquitectura del sistema.....	46
Figura 4.14: Modelo de clases del sistema.....	47
Figura 4.15: Diagrama dominio/diseño del sistema.....	49
Figura 4.16: Diagrama de capas del sistema	50
Figura 4.17: Ejemplo primera propuesta para escoger pedazos	55
Figura 4.18: Ejemplo segunda propuesta para escoger pedazos	55
Figura 4.19: Ejemplo de un escudo con pedazos sin información	58
Figura 5.1: Página de bienvenida.....	63
Figura 5.2: Formulario de registro.....	64
Figura 5.3: Inicio de sesión.....	65
Figura 5.4: Apartado de estadísticas	65
Figura 5.5: Perfil del usuario.....	66
Figura 5.6: Edición del perfil.....	66
Figura 5.7: Cambio de contraseña.....	67
Figura 5.8: Seleccionar modo de juego	67
Figura 5.9: Partida en solitario comenzada	68
Figura 5.10: Ejemplo de envío de un intento	69
Figura 5.11: Ejemplo de fallo en solitario.....	69
Figura 5.12: Victoria en solitario	70
Figura 5.13: Derrota en solitario	70
Figura 5.14: Crear partida uno contra uno local.....	71
Figura 5.15: Partida uno contra uno local	72
Figura 5.16: Fallo en partida uno contra uno local.....	72
Figura 5.17: Acierto en partida uno contra uno local.....	73
Figura 5.18: Resultado partida uno contra uno local	74
Figura 5.19: Creación partida en línea	75
Figura 5.20: Unirse a partida en línea	76
Figura 5.21: Lobby de partidas en línea	77
Figura 5.22: Partida en línea.....	77
Figura 5.23: Acierto en partida en línea	78
Figura 5.24: Fallo en partida en línea	78
Figura 5.25: Minijuego esperando terminar partida en línea	79
Figura 5.26: Resultado partida en línea.....	80
Figura 5.27: Resultado por abandono en línea	80

Figura 5.28: Pagina de bienvenida para administradores	81
Figura 5.29: Opciones módulo de administración.....	81
Figura 5.30: Listado de usuario para el administrador.....	82
Figura 5.31: Perfil de un usuario desde la vista del administrador	82
Figura 5.32: Listado de imágenes para el administrador	83
Figura 5.33: Detalles de una imagen	84
Figura 5.34: Formulario para añadir una nueva imagen	85
Figura 5.35: Estadísticas generales de las imágenes	85
Figura 6.1: Escudo con pedazos no validos	88
Figura 6.2: Ejemplo registro de Git Hub Actions	91
Figura 9.1: Ejemplo de escudos retro.....	100
Figura 9.2: Gráfica de barras con los aciertos por país.....	101
Figura 9.3: Ratio de acierto por género.....	102
Figura 9.4: Ratio de acierto si el escudo incluye el año.....	103
Figura 9.5: Ratio de acierto si el escudo incluye las iniciales del club.....	103
Figura 9.6: Ratio de acierto si el escudo incluye el nombre.....	104
Figura 9.7: Top 10 equipos con más apariciones.....	104

Capítulo 1 Definición de objetivos

El desarrollo de este Trabajo Fin de grado tiene varios objetivos siendo uno de ellos el objetivo principal y el resto una consecuencia de este mismo.

El objetivo principal es crear una aplicación web con una interfaz vistosa y llamativa para jugar al juego Con Un Pedazo Es Suficiente, un juego que consiste en mostrar un pedazo de un escudo de un equipo de fútbol y que el usuario que esté jugando, tenga que adivinar a qué equipo pertenece esa porción de escudo. Si el jugador no consigue acertar, se mostrará un pedazo diferente hasta que acierte o utilice todos sus intentos.

Además, para aprovechar esta funcionalidad de la aplicación habrá dos modos de juego extra, en uno de los modos dos jugadores puedan enfrentarse desde el mismo dispositivo para tratar de adivinar un escudo cada uno y competir entre ellos. En el otro modo de juego se enfrentarán dos jugadores en línea y cada uno tendrá que adivinar tres escudos.

La intención es que el modo de juego individual y el uno contra uno local sirvan como modos de práctica y que el modo competitivo online de lugar a una clasificación o a un conjunto de estadísticas, para incentivar a los usuarios a que jueguen más y conozcan más equipos para poder liderar la clasificación.

Uno de los objetivos secundarios de este trabajo es crear una aplicación web desde cero y aprender a seleccionar las herramientas necesarias para poder llevar a cabo el desarrollo, al contrario que en todos los trabajos realizados a lo largo del grado universitario donde en todos los proyectos se partía un proyecto base con unas tecnologías predeterminadas para que trabajar sobre él.

Otro de los objetivos secundarios es dotar a la aplicación de un módulo de administración a través del cual los administradores puedan tener información sobre las partidas, las imágenes y los jugadores. Además, para balancear el juego, los administradores podrán deshabilitar jugadores, añadir nuevas imágenes y controlar cuáles pueden salir en las partidas.

Finalmente se busca recopilar datos de las partidas para poder hacer un estudio de las estadísticas, sacando conclusiones sobre los clubes y ligas más conocidos, el grado de conocimiento de los usuarios sobre clubes femeninos o la influencia que pueden tener los escudos que tienen años de fundación, iniciales del club o incluso el nombre.

Capítulo 2 Análisis de antecedentes y aportación realizada

En los últimos años los juegos de tipo quiz han tenido un gran auge a través de internet y por esto cada vez podemos encontrar más aplicaciones de este estilo en las que tenemos que adivinar algo de una temática concreta a partir de una serie de pistas o preguntas.

Uno de los ejemplos más famosos de internet es la conocida aplicación Wordle (<https://wordlegame.org/es>). Esta aplicación nos presenta el desafío diario de adivinar una palabra y para darnos pistas, cuando probamos con una palabra, las letras de la palabra que hemos introducido toman distintos colores siguiendo estas reglas, si una letra toma el color gris, esa letra no pertenece a la palabra del día. Si el color que toma es amarillo, la letra pertenece a la palabra del día, pero no está en la posición correcta. En cambio, si el color que toma la letra es verde, no solo pertenece a la palabra del día, sino que también está en la posición correcta. Y a partir de esta información que vamos obteniendo tras cada intento, tenemos que intentar adivinar la palabra del día.

Un concepto distinto es el de Flagle (<https://www.flagle.io/>), otra aplicación del estilo quiz donde cada día se puede jugar una vez y hay que adivinar la bandera de un país. El juego consta de 6 seis cuadros, cada uno de ellos es un trozo de la misma bandera y nosotros tenemos que tratar de adivinar de qué país se trata. En primer lugar, la bandera está completamente oculta y los cuadros solo tienen un color gris, por cada intento que hagamos, si fallamos se nos mostrará un cuadro más. Además, como en cada intento, tenemos que probar seleccionando un país, como información adicional, la aplicación nos dirá cuál es la distancia entre el país al que corresponde la bandera y el país que nosotros hemos seleccionado, aunque esto sea una pista un poco más secundaria. Una de las grandes diferencias entre Wordle y Flagle es que Flagle utiliza las imágenes para dar pistas y Wordle tan solo utiliza letras y colores.

Este formato de juego se ha extendido tanto que hay quiz de todo tipo de temáticas, otra temática muy recurrente es la música, donde podemos encontrar aplicaciones como Heardle Unlimited (<https://heardleunlimited.io/>). Una aplicación donde tendremos que adivinar una canción y para esta labor iremos escuchando fragmentos cada vez más grandes de esta misma. Hablando de la temática musical cabe destacar que esta temática ha hecho que los juegos quiz se extiendan por redes sociales, solo tendríamos que buscar “Guess the song” o “Adivina la canción” en Youtube (<https://www.youtube.com/>) o Tik Tok (<https://www.tiktok.com/es/>) y podremos

ver una gran cantidad de videos que siguen el formato de mostrarnos un fragmento de una canción para a partir de dicho fragmento la adivinemos.

Uno de los aspectos fundamentales a la hora de trabajar en la aplicación “Con Un Pedazo Es Suficiente” era definir que temática se utilizaría y para ello surgieron varias alternativas como podrían ser imágenes de lugares importantes de Sevilla, escudos de equipos deportivos o personajes famosos, pero por diferentes factores que explicaremos más adelante decidí optar por los escudos de equipos deportivos, concretamente de clubes de fútbol con tal de acotar un poco la dificultad para los usuarios.

Una vez establecida la temática de la aplicación, es el momento de ver las distintas alternativas existentes que utilicen esta temática. En este caso, hay una página web que destaca por encima de todas por su gran cantidad de modos de juego y por la gran cantidad de usuarios que juegan en ella, estamos hablando de “Play Football Games – Football Quiz Games” (<https://playfootball.games/>). Dentro de la aplicación destacan los siguientes modos de juego:

- Who Are Ya? Un juego donde a partir de pistas como la edad, el dorsal, la posición, el club y la nacionalidad, diariamente tendremos que adivinar un jugador de la liga que seleccionemos. Para obtener pistas, tenemos que seleccionar un jugador y si el jugador seleccionado coincide en alguna de las pistas con el jugador del día, el recuadro de la pista que coincide se coloreará de color verde, en el caso de que no coincida una pista, se coloreará de color gris. En pistas como el dorsal o la edad, si no coincide, se nos aportará información que nos indicará si la edad o el dorsal del jugador del día es mayor o menor que la del jugador que hemos seleccionado. Hay dos posibilidades para este modo de juego y es que podemos escoger entre tener la imagen del jugador durante la partida o no tenerla, si pedimos que se nos muestre la foto, esta estará pixelada y será prácticamente imposible reconocerlo por lo que esta pista no nos aportara demasiada información.
- Football Wordle, una versión de Wordle con temática de futbol, siguiendo las mismas reglas que la versión original donde diariamente hay que adivinar el nombre de un jugador de fútbol.
- Guess The Football Club, este juego sigue la temática de "Who Are Ya?", pero en vez de adivinar un jugador, en este caso tendremos que adivinar un club, para adivinar el club, tendremos acceso a pistas como la

nacionalidad, el año de fundación, la capacidad de su estadio y la distancia entre el club del día y el club que nosotros escojamos. Al igual que el modo “Who Are Ya?”, tendremos la opción de que se nos muestre la imagen del club pixelada o que no se nos muestre, con la diferencia de que en este caso si pedimos que nos muestre la imagen pixelada, en el centro del escudo se nos mostrará una pequeñísima porción sin pixelar que con cada intento que fallemos ira creciendo.

Una vez estudiadas varias aplicaciones con un concepto parecido, llega el momento de ver a grandes rasgos cuáles serán las características que diferencien mi proyecto de todos los antecedentes.

En primer lugar, CUPES a diferencia de las aplicaciones citadas que utilizan pistas sonoras o conceptuales, utilizará tan solo pistas visuales como son los pedazos de los escudos. Es cierto que algunos de los modos de juego citados en los antecedentes se utilizaban imágenes como pistas suplementarias, pero siempre se utilizan imágenes pixeladas y en cada ronda pasan a ser un poco más nítidas, la diferencia con nuestro proyecto estaría en que nosotros usaremos siempre imágenes nítidas que serán pedazos escogidos de forma aleatoria siempre y cuando cumplan unos requisitos mínimos de información para que sean pistas validas.

En segundo lugar, en la mayoría de las aplicaciones del estilo, se puede jugar una partida al día ya que la finalidad de las aplicaciones es adivinar la palabra, el jugador, el escudo, la bandera o la canción del día y una vez que la adivinas o que gastas todos tus intentos, no puedes volver a jugar hasta el día siguiente. Sin embargo, en CUPES se pueden jugar tantas partidas como se quiera sin ninguna limitación.

Otro atractivo de nuestro proyecto es que en ninguna de las aplicaciones de este ámbito existe un modo de juego en el que dos usuarios puedan enfrentarse desde el mismo dispositivo y competir para ver quien es más rápido adivinando y no solo eso, sino que en “Con Un Pedazo Es Suficiente” además habrá un modo de juego donde dos jugadores aleatorios cualesquiera puedan enfrentarse en una partida donde cada uno tenga que adivinar un número de escudos y gane quien más escudos acierte o quien menos tiempo tarde en caso de empate en el número de intentos.

Adicionalmente la aplicación a desarrollar tendrá un apartado estadístico que incluirá un ranking, esto motivará a los jugadores a jugar más ya que sus estadísticas quedarán ligadas a sus cuentas y aspirar a liderar el ranking de jugadores con más victorias, a diferencia de las partidas anteriormente mencionadas donde los jugadores no necesitan tener una cuenta y por lo tanto sus estadísticas no quedan ligados a ellos

por lo que los usuarios no pueden acceder a ningún apartado que contenga sus estadísticas.

Por otro lado, la aplicación será provista de un módulo de administración donde los administradores podrán gestionar las imágenes y los jugadores o ver más estadísticas para poder balancear la dificultad del juego.

Finalmente, en la aplicación 'Play Football Games – Football Quiz Games', cada modo de juego cuenta con una versión específica para cada liga. Por ejemplo, al jugar en el modo 'Who Are Ya?', se debe seleccionar previamente una liga para que el jugador a adivinar pertenezca a dicha liga. No obstante, en nuestra aplicación para aumentar la dificultad se ha decidido que en nuestra aplicación pueda aparecer cualquiera de los escudos habilitados. Esto implica que los escudos de cada partida serán seleccionados de un conjunto de alrededor de 127 escudos. Estos clubes pueden representar tanto el fútbol masculino como el femenino, pertenecer a primeras o segundas divisiones, y provenir de diversos países.

Capítulo 3 Análisis temporal y de costes de desarrollo

A lo largo de esta sección analizaremos las distintas tareas realizadas para diseñar e implementar la aplicación web y desarrollar este Trabajo Fin de Grado. Para cada tarea se aportará una descripción con la que se pueda comprender todo lo que conlleva cada tarea y todo el trabajo que hay detrás de cada una. Además, se incluirá para cada tarea una estimación inicial del tiempo necesario para completar la tarea, realizada antes de completar esta misma, por otro lado, para poder comparar como de buenas fueron las estimaciones, se podrá comparar la estimación con el tiempo real que se le ha acabado dedicando a cada tarea.

Como no todas las tareas son iguales, y no todas requieren las mismas capacidades para ser completadas, junto a cada tarea se adjuntará el rol que habría que desempeñar para completarla. A partir de estos últimos datos, podremos hacer un análisis de los costes más detallado, ya que así podremos tener en cuenta precio por hora que habría que pagar en función del rol que se desempeña. Esta opción es más realista y detallada que tomar la cantidad de horas total necesarias para completar el proyecto al completo y multiplicarla por un precio.

Listado de tareas a realizar:

- **Título:** Selección de tecnologías e inicialización del proyecto.
 - **Descripción:** Esta es una tarea crucial para el desarrollo del proyecto ya que de las herramientas seleccionadas dependerá parte del éxito de este mismo. Para tomar esta decisión hay que tener en cuenta muchos factores, siendo el factor principal los requisitos y teniendo otros factores importantes como podrían ser los conocimientos del desarrollador en las distintas tecnologías utilizadas. Seleccionar las tecnologías no solo implica escoger un lenguaje de programación y un framework, sino que también implica prever las distintas librerías externas que podrías ser útiles en el ciclo de vida del proyecto, así como las incompatibilidades que pueden existir entre estas. También habría que tener en cuenta para esta tarea, el tiempo necesario para aprender algunas de las tecnologías que se usarán en el proyecto.
 - **Rol:** El rol necesario para llevar a cabo esta tarea es el de jefe de proyecto. Ya que quien tenga este rol es quien debe tomar este tipo de decisiones cruciales que afectan a la totalidad del proyecto, porque es el quien tiene la visión más general del proyecto.

- **Tiempo estimado:** 20 horas.
- **Título:** Gestión de la base de datos.
 - **Descripción:** La gestión de la base de datos conlleva generar datos para inicializar la aplicación y que esta funcione correctamente. En el caso de esta aplicación, principalmente la tarea se resume en buscar imágenes de escudos con el tamaño y formato necesarios para que la aplicación funcione correctamente. En el caso de no encontrar imágenes que cumplan los requisitos de tamaño y formato, se utilizarán imágenes que serán modificadas para que finalmente cumplan con los requisitos. Además, habrá que generar un script para que la base de datos de la aplicación pueda tener acceso a dichas imágenes.
 - **Rol:** Por lo tanto, esta tarea deberá ser desarrollada por un administrador de base de datos que es quien conocerá mejor la forma de generar los datos iniciales para que la aplicación funcione correctamente.
 - **Tiempo estimado:** 10 horas.
- **Título:** Redacción de la memoria.
 - **Descripción:** El fin de esta labor es redactar todos los apartados de la memoria de forma progresiva, según se puedan ir completando. Otra de las implicaciones de esta tarea es auditar constantemente la memoria para comprobar que cumple con los estándares de calidad. Completar la tarea lleva consigo tareas implícitas como hacer una planificación estimando los tiempos del proyecto, analizar los requisitos, realizar un diseño, documentar decisiones tomadas etc.
 - **Rol:** Por esto será una tarea que realizará el jefe de proyecto, ya que es la persona del proyecto que tiene una visión más global de este mismo.
 - **Tiempo estimado:** 80 horas.
- **Título:** Diseño de la interfaz web.
 - **Descripción:** La finalidad que se busca obtener con esta tarea es conseguir diseñar una interfaz de usuario que sea bonita y llamativa, además de que la interfaz se vea bien tanto en dispositivos móviles como en pantallas de mayor tamaño.
 - **Rol:** Para esta labor, se necesitará un desarrollador Frontend.

- **Tiempo estimado:** 30 horas.
- **Título:** Desarrollo del modo de juego individual.
 - **Descripción:** esta tarea consiste en desarrollar un modo de juego para un solo jugador, donde el jugador tenga que adivinar un escudo con una cantidad finita de intentos, perdiendo la partida si no consigue adivinar el escudo en cuatro intentos o menos.
 - **Rol:** Esta tarea no solo requiere de desarrollo a nivel de Backend, sino que también requiere implementar parte de la lógica por la cual se mostrarán los pedazos seleccionados, por lo que se necesitará a un desarrollador Fullstack para esta labor.
 - **Tiempo estimado:** 10 horas.
- **Título:** Desarrollo del modo de juego uno contra uno local.
 - **Descripción:** Al igual que la anterior tarea, esta consta de desarrollar un modo de juego, pero en este caso, el modo de juego será para que puedan jugar dos jugadores desde el mismo equipo, la idea es que cada jugador tenga que adivinar un escudo y que gane quien menos intentos necesite.
 - **Rol:** En este caso, al igual que en el anterior, la tarea deberá ser desarrollada por un desarrollador Fullstack.
 - **Tiempo estimado:** 10 horas.
- **Título:** Desarrollo del modo de juego competitivo.
 - **Descripción:** Otra tarea que se basa en desarrollar un modo de juego, pero esta vez será un modo de juego en el que dos jugadores puedan competir a través de internet, para ver quién consigue adivinar tres escudos necesitando menos intentos.
 - **Rol:** Para esta tarea también se necesitará un desarrollador Fullstack.
 - **Tiempo estimado:** 15 horas.
- **Título:** Configuración de seguridad de la aplicación.
 - **Descripción:** El objetivo de esta tarea es proveer de seguridad a la web, en primer lugar, esto trataría de controlar que los usuarios no accedan a páginas a las que no deben acceder, como podría ser acceder a una partida que no es del usuario. También hay que dotar a la aplicación con un sistema de inicio

de sesión seguro para los usuarios, junto con un sistema de registro donde las contraseñas de nuestros usuarios queden guardas de forma segura siendo encriptadas. Esta labor agrupa otras muchas responsabilidades como podría ser proteger la aplicación de ataques como SQLInjections, ScriptInjectios, ataques CSRF, etc.

- **Rol:** Para esta tarea se necesitará un experto en seguridad informática.
- **Tiempo estimado:** 15 horas.
- **Título:** Refactorización de código.
 - **Descripción:** La refactorización de código es un proceso esencial en el desarrollo software, ya que tiene como objetivo mejorar la calidad, legibilidad y mantenibilidad del código, el efecto de esta tarea se verá en la cantidad de código que se podrá reutilizar entre los distintos modos de juego, en vez de tener muchísima cantidad de código repetido.
 - **Rol:** Esta tarea será desarrollada por un desarrollador Backend.
 - **Tiempo estimado:** 15 horas.
- **Título:** Protocolo de selección de pedazos.
 - **Descripción:** Esta tarea consta de desarrollar un método para comprobar que los pedazos de imágenes que se muestran contienen pixeles de más de un color, y, por lo tanto, se comprueba que aportan información suficiente para poder averiguar el escudo.
 - **Rol:** Para esta tarea se necesitará un desarrollador Backend.
 - **Tiempo estimado:** 5 horas.
- **Título:** Sistema de abandono de partidas.
 - **Descripción:** esta tarea consta de proveer a la aplicación de un sistema de abandono de partidas de cara al modo de juego en línea, para que, si un jugador no conoce los escudos que le aparecen y desea abandonar para no perder su tiempo, pueda abandonar la partida, dándose esta por perdida. Además, se deberá implementar un sistema para que, si un jugador que entra en una partida en línea y tarda más de 7 minutos en terminar, se le dé por abandonada para evitar así que el otro jugador espere eternamente hasta que el primero termine.

- **Rol:** Para esta tarea se necesitará un desarrollador Backend.
 - **Tiempo estimado:** 3 horas.
- **Título:** Reuniones.
 - **Descripción:** Las reuniones juegan un papel fundamental en el desarrollo del proyecto ya que en ellas se puede mostrar al cliente (tutor del TFG) los avances obtenidos, para buscar correcciones y mejoras.
 - **Rol:** Esta tarea será llevada a cabo por el jefe del proyecto.
 - **Tiempo estimado:** 10 horas.
- **Título:** Crear módulo de estadísticas.
 - **Descripción:** Para incentivar a los jugadores a jugar más partidas en los distintos modos de juego, estos deben tener acceso a una serie de estadísticas y sobre todo a un ranking con los mejores jugadores. Además, el administrador también contará con estadísticas detalladas sobre las imágenes. Esto permitirá al administrador identificar la procedencia de los equipos más conocidos, determinar las divisiones más populares, y analizar si los equipos de fútbol masculino o femenino tienen de mayor reconocimiento.
 - **Rol:** Esta tarea será llevada a cabo por un desarrollador Backend.
 - **Tiempo estimado:** 10 horas.
- **Título:** Módulo de administración.
 - **Descripción:** Una vez la aplicación ya esté completa y desplegada, los administradores deberían tener el poder de cambiar las imágenes de los modos de juego deshabilitando las ya existentes y añadiendo nuevas imágenes, además, deberá poder habilitar o deshabilitar a los jugadores registrados.
 - **Rol:** Esta tarea se basará en desarrollar la funcionalidad de dicho módulo, por lo que esta tarea será llevada a cabo por un desarrollador Backend.
 - **Tiempo estimado:** 10 horas.

- **Título:** Pruebas.
 - **Descripción:** Esta tarea es fundamental para asegurarnos de que la aplicación funciona correctamente y que su comportamiento no cambia con el paso del tiempo y el desarrollo de nuevas funcionalidades.
 - **Rol:** Las pruebas serán desarrolladas por un tester de calidad.
 - **Tiempo estimado:** 30 horas.
- **Título:** Despliegue.
 - **Descripción:** Para que los usuarios puedan jugar hay que encontrar una plataforma donde desplegar la aplicación y además habrá que automatizar el proceso de despliegue para que siempre esté disponible la última versión de la aplicación.
 - **Rol:** Esta tarea será realizada por un DevOps.
 - **Tiempo estimado:** 7 horas.
- **Título:** Análisis de datos.
 - **Descripción:** Una vez que la aplicación esté lista y sea desplegada, se intentará conseguir usuarios pilotos para que la prueben y a partir de los datos de estos usuarios pilotos se realizará un análisis de las estadísticas de la aplicación para poder sacar algunas conclusiones.
 - **Rol:** Esta tarea será realizada por analista de datos.
 - **Tiempo estimado:** 20 horas.

Tarea	Tiempo estimado	Tiempo real dedicado	Rol desempeñado
Selección de tecnologías e inicialización del proyecto	20 horas	21 horas	Jefe de proyecto
Gestión de la base de datos	10 horas	10 horas	Administrador de base de datos
Redacción de la memoria	80 horas	80 horas	Jefe de proyecto
Diseño de la interfaz de web	30 horas	34 horas	Desarrollador Frontend
Desarrollo del modo de juego individual	10 horas	10 horas	Desarrollador Fullstack

Desarrollo del modo de juego uno contra uno local	10 horas	11 horas	Desarrollador Fullstack
Desarrollo del modo de juego competitivo	15 horas	15 horas	Desarrollador Fullstack
Configuración de la seguridad de la aplicación	15 horas	15 horas	Experto en seguridad informática
Refactorización de código	15 horas	15 horas	Desarrollador Backend
Protocolo de selección de pedazos	5 horas	5 horas	Desarrollador Backend
Sistema de abandono de partidas	3 horas	3 horas	Desarrollador Backend
Crear módulo de estadísticas	10 horas	10 horas	Desarrollador Backend
Módulo de administración	10 horas	12 horas	Desarrollador Backend
Pruebas	30 horas	32 horas	Tester de calidad
Reuniones	10 horas	10 horas	Jefe de proyecto
Despliegue	7 horas	7 horas	Dev Ops
Análisis de datos	20 horas	20 horas	Analista de datos
Total	300 horas	310 horas	

Cuadro 3.1:Resumen de tiempo por tarea

En la anterior tabla se puede ver la diferencia entre el tiempo estimado para cada tarea y el tiempo que finalmente se ha dedicado a cada una de ellas, como se puede ver, no ha habido demasiadas desviaciones, una prueba de ello es que el tiempo total estimado era de trescientas horas y el tiempo que realmente se ha empleado ha sido de trescientas diez horas. Aún así en varias tareas ha habido pequeñas desviaciones, siendo las más significativas la desviación de cuatro horas en el diseño de la interfaz, que se ha debido a unas mejoras para hacer la interfaz más responsive y la desviación en las pruebas ya que al hacer cambios en la interfaz, se han tenido que modificar las pruebas de interfaz que se habían implementado.

Una vez realizado el análisis temporal del proyecto, es momento de utilizar los registros de tiempo empleado en las distintas tareas para calcular un presupuesto. Hay que prestar a atención a que cada tarea es desarrollada por un rol específico y que cada

rol tiene un sueldo distinto, lo que será tomado en cuenta para calcular el coste del proyecto. En la siguiente tabla se muestra el tiempo que se ha empleado en cada rol y el coste de los roles.

Rol	Tiempo	Coste	Coste por Rol
Jefe de proyecto	111 horas	35,21€/h	3.908,31€
Desarrollador Backend	45 horas	21,95€/h	987,75€
Desarrollador Frontend	34 horas	21,95€/h	746,3€
Desarrollador Fullstack	36 horas	21,95€/h	790,2€
Experto en seguridad informática	15 horas	18,28€/h	274,2€
Administrador de la base de datos	10 horas	26,83€/h	268,3€
DevOps	7 horas	21,95€/h	153,65€
Tester de calidad	32 horas	19,8€/h	633,6€
Analista de datos	20 horas	21,95€/h	439€
Total	290		8.201,31€

Cuadro 3.2: Costes del proyecto

Los precios han sido tomados del documento “PERFILES PROFESIONALES ÁMBITO INFORMÁTICO” de la junta de Andalucía https://tfc.eii.us.es/TFG/APP/connector/0/33/href/informe_precios_perfiles_informaticos.pdf, teniendo en cuenta el 21% de IVA y tomando para todos los casos el precio mínimo por hora de un perfil Junior.

Capítulo 4 Análisis de requisitos, diseño e implementación

Para gestionar los requisitos del proyecto se utilizará la aplicación REMUS <https://github.com/amador-duran-toro/remus> desarrollada por el profesor de la universidad Amador Durán Toro.

Historias de usuario

HU-0001	Jugar partidas
Versión	1.0 (15/10/2023)
Descripción	Como usuario quiero jugar a los distintos modos de juego de la aplicación para pasar el rato.
Comentarios	Ninguno

Cuadro 4.1: Historia de usuario jugar partidas

HU-0002	Consultar estadísticas
Versión	1.0 (15/10/2023)
Descripción	Como usuario quiero poder consultar estadísticas de los diferentes modos de juego para motivarme a jugar más partidas.
Comentarios	Ninguno

Cuadro 4.2: Historia de usuario consultar estadísticas

HU-0003	Registro de usuarios
Versión	1.0 (15/10/2023)
Descripción	Como usuario quiero poder registrarme para jugar partidas en línea y que el sistema almacene mis estadísticas.
Comentarios	Ninguno

Cuadro 4.3: Historia de usuario registro de usuarios

 HU-0004	Consultar el manual
Versión	1.0 (15/10/2023)
Descripción	Como usuario quiero consultar un manual para aprender a jugar a los distintos modos de juego.
Comentarios	Ninguno

Cuadro 4.4: Historia de usuario consultar el manual

 HU-0005	Administración de imágenes
Versión	1.0 (18/11/2023)
Descripción	Como administrador quiero añadir y eliminar imágenes para tener más variedad y un juego más balanceado.
Comentarios	Ninguno

Cuadro 4.5: Historia de usuario administración de imágenes

 HU-0006	Administración de jugadores
Versión	1.0 (18/11/2023)
Descripción	Como administrador quiero gestionar los usuarios registrados para castigar a quienes tengan malos comportamientos en las partidas en línea
Comentarios	Ninguno

Cuadro 4.6: Historia de usuario administración de jugadores

 HU-0007	Estudio de estadísticas
Versión	1.0 (24/01/2024)
Descripción	Como administrador quiero tener acceso a distintos datos sobre las partidas para poder estudiar en profundidad los resultados de las mismas.
Comentarios	Ninguno

Cuadro 4.7: Historia de usuario estudio de estadísticas

Requisitos generales

 OBJ-0001	Juego solitario
Versión	1.0 (15/10/2023)
Descripción	El sistema deberá permitir que todos los usuarios puedan jugar partidas en el modo de juego para un jugador.
Comentarios	Ninguno

Cuadro 4.8: Requisito general juego solitario

 OBJ-0002	Juego local
Versión	1.0 (15/10/2023)
Descripción	El sistema deberá permitir que todos los usuarios puedan jugar partidas uno contra uno desde el mismo dispositivo.
Comentarios	Ninguno

Cuadro 4.9: Requisito general juego local

 OBJ-0003	Juego en línea
Versión	1.0 (15/10/2023)
Descripción	El sistema deberá permitir que todos los usuarios registrados puedan jugar partidas uno contra uno en línea con rivales aleatorios.
Comentarios	Ninguno

Cuadro 4.10: Requisito general juego en línea

 OBJ-0004	Gestión de usuarios
Versión	1.0 (15/10/2023)
Descripción	El sistema deberá permitir que cualquier usuario pueda registrarse e iniciar sesión.
Comentarios	Ninguno

Cuadro 4.11: Requisito general gestión de usuarios

 OBJ-0005	Módulo de administración
Versión	1.0 (18/11/2023)
Descripción	El sistema deberá permitir que los administradores gestionen las imágenes y los jugadores.
Comentarios	Ninguno

Cuadro 4.12: Requisito general módulo de administración

 OBJ-0006	Exportación de estadísticas
Versión	1.0 (24/01/2024)
Descripción	El sistema deberá permitir que los administradores exporten las estadísticas de las partidas en solitario según varios criterios.
Comentarios	Ninguno

Cuadro 4.13: Requisito general exportación de estadísticas

Casos de uso

Diagramas de casos de uso

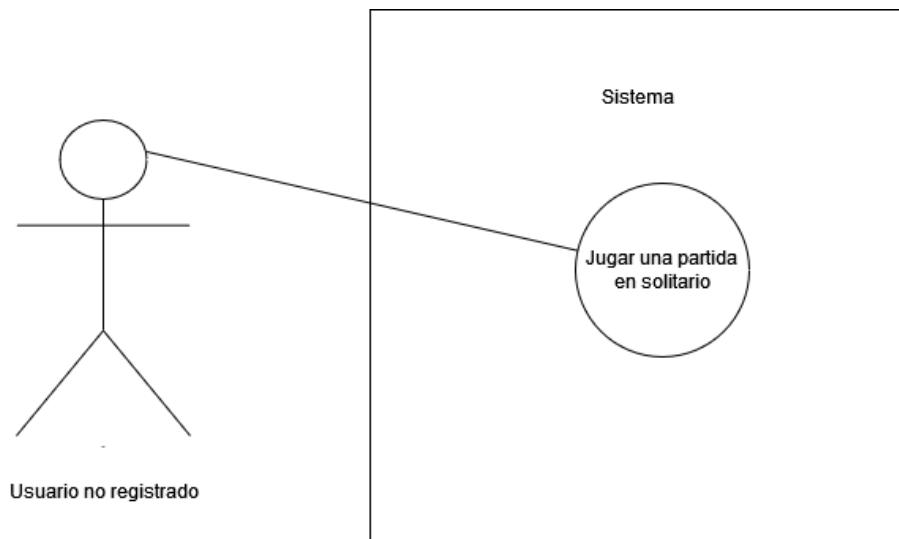


Figura 4.1: Diagrama de caso de uso jugar en solitario

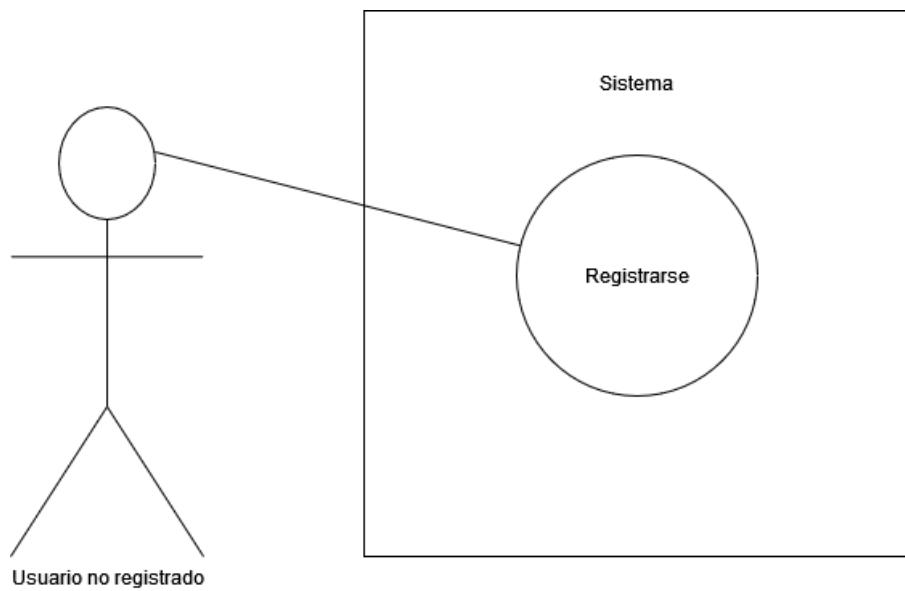


Figura 4.2: Diagrama de caso de uso registrarse

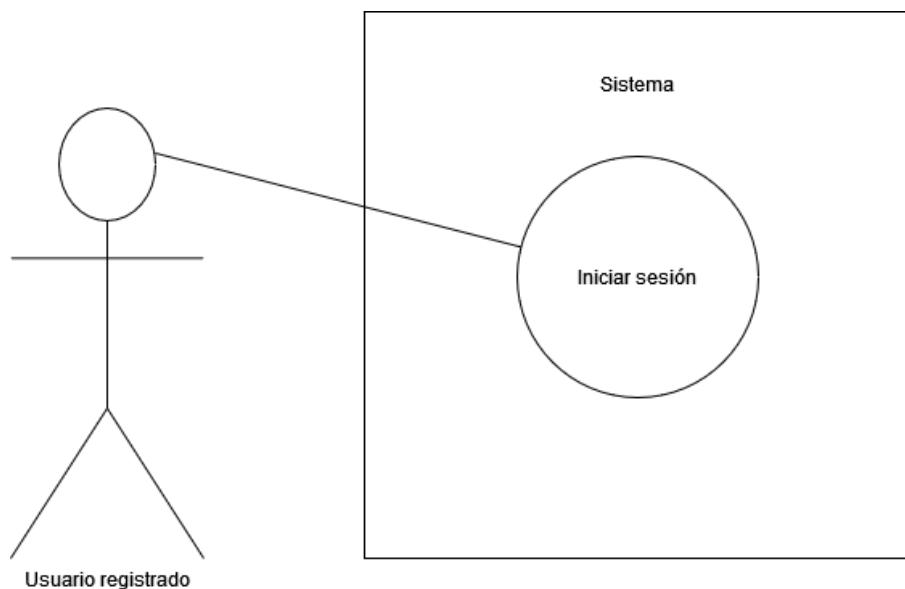


Figura 4.3: Diagrama de caso de uso iniciar sesión

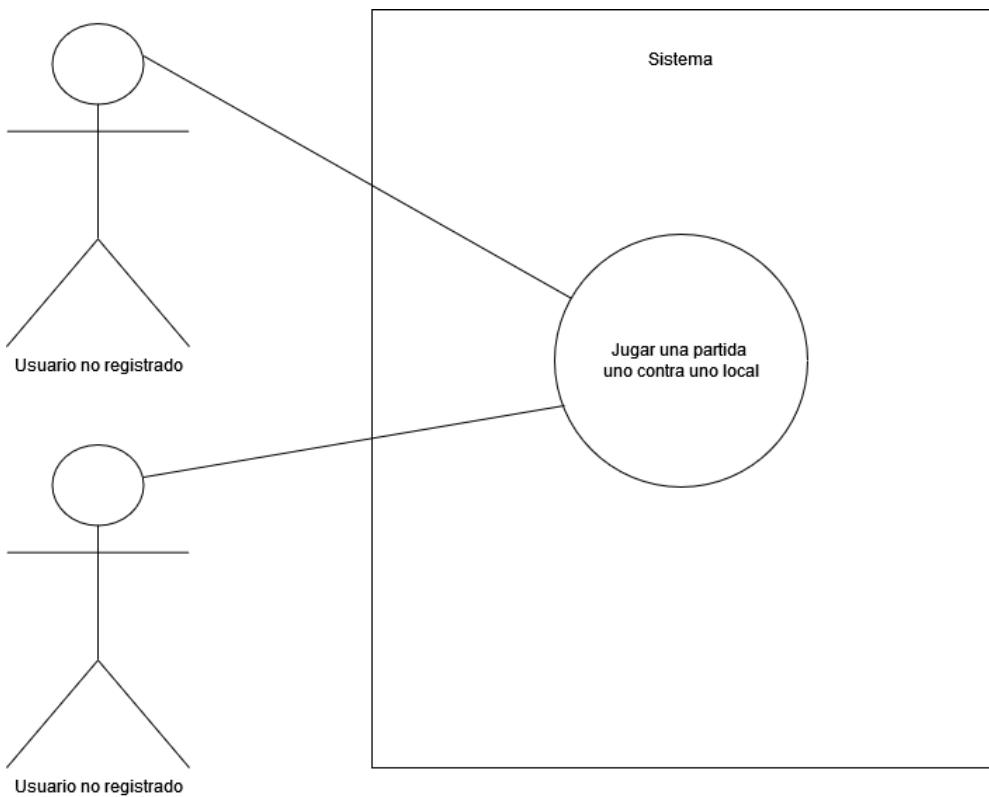


Figura 4.4: Diagrama de caso de uso jugar uno contra uno local

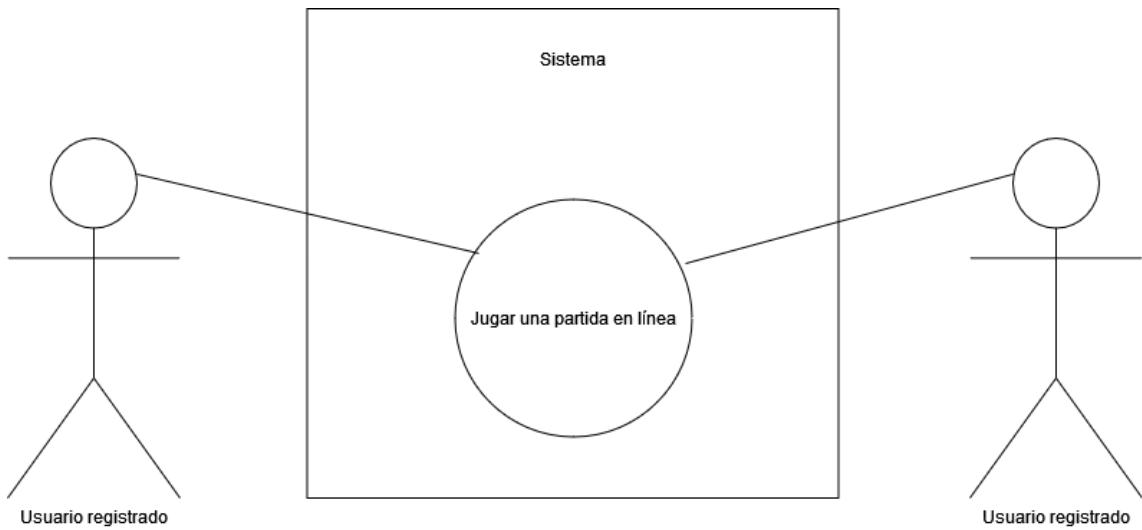


Figura 4.5: Diagrama de caso de uso jugar partida en línea

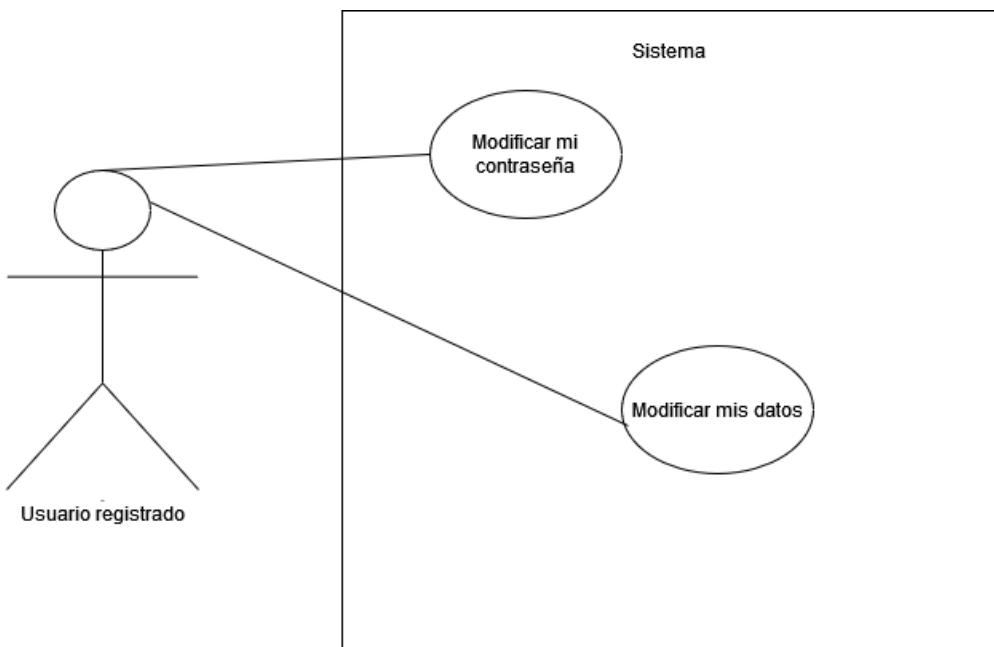


Figura 4.6: Diagrama de caso de uso editar usuario

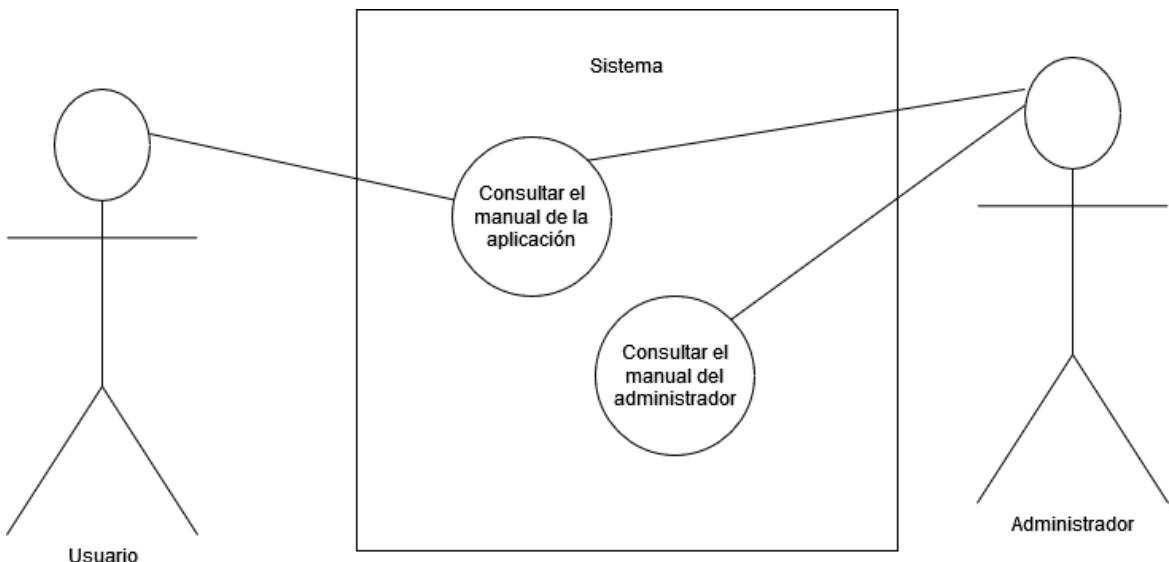


Figura 4.7: Diagrama de caso de uso consultar el manual

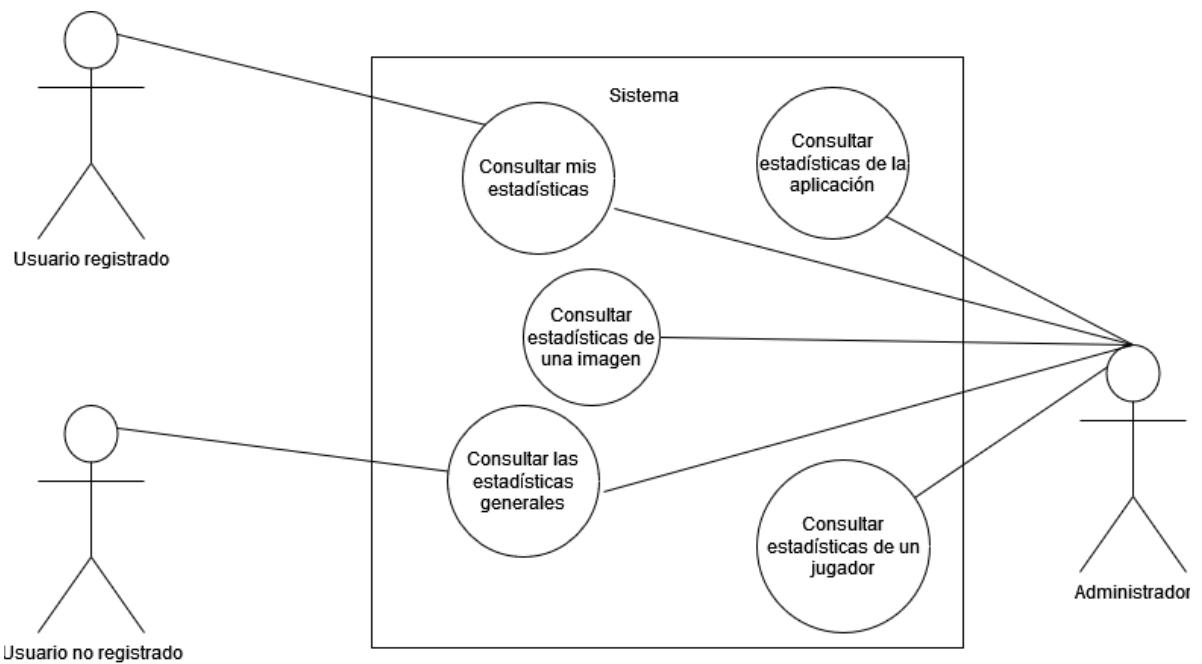


Figura 4.8: Diagrama de caso de uso consultar estadísticas

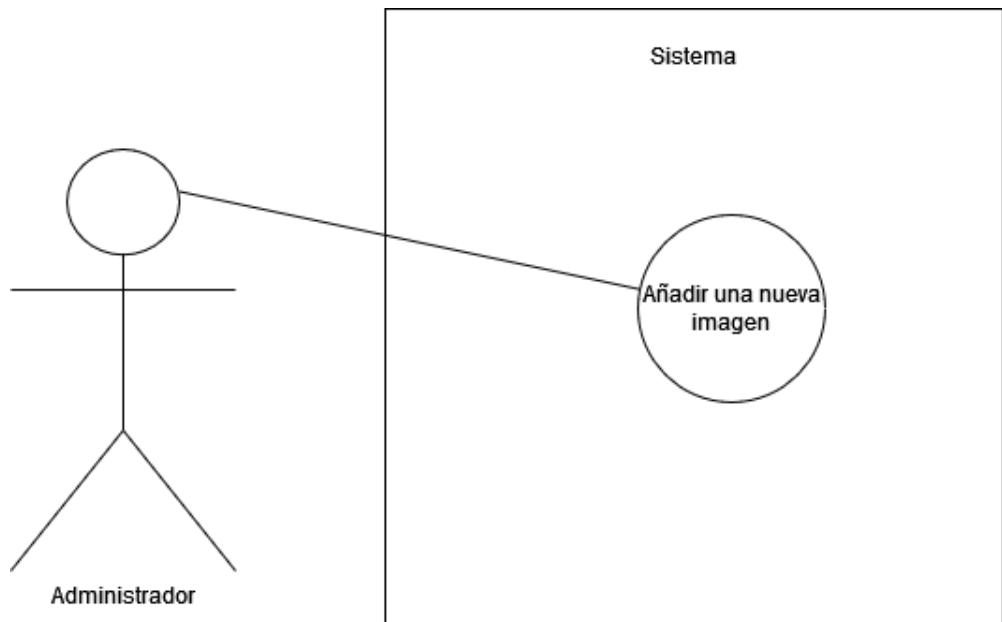


Figura 4.9: Diagrama de caso de uso añadir nueva imagen

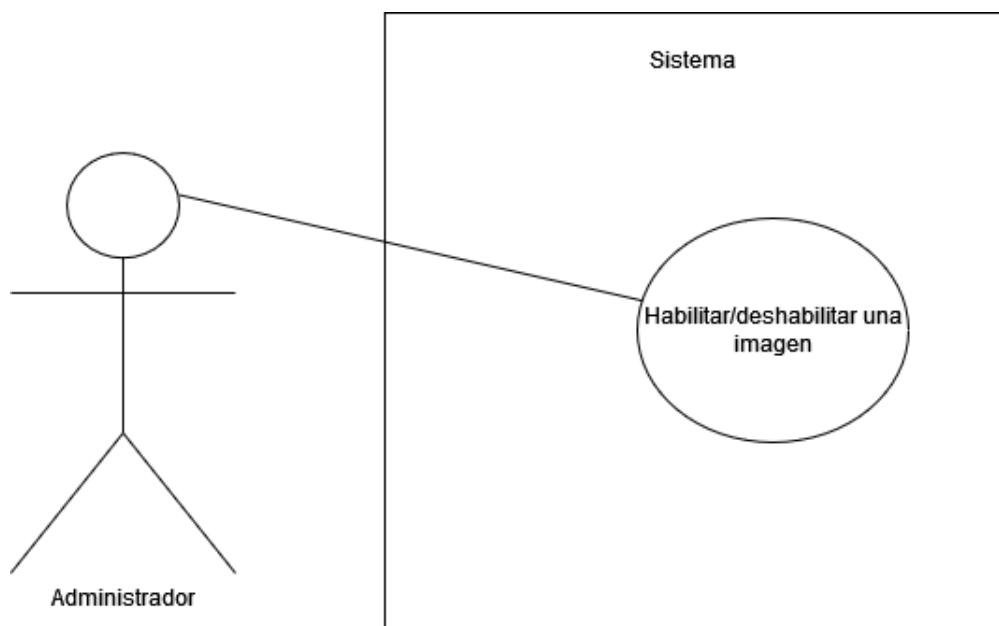


Figura 4.10: Diagrama de caso de uso habilitar/deshabilitar imagen

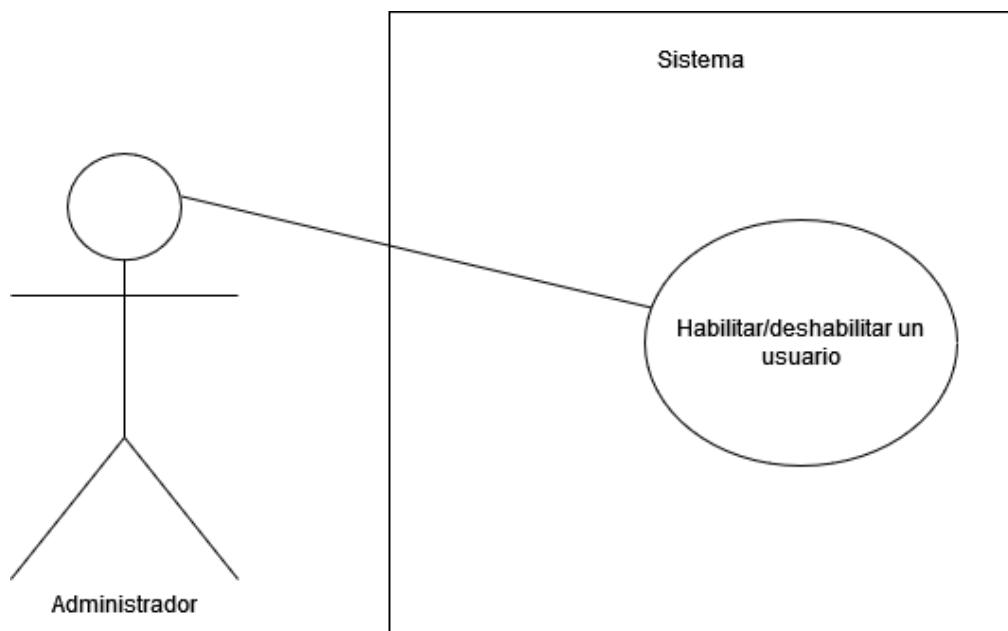


Figura 4.11: Diagrama de caso de uso habilitar/deshabilitar usuario

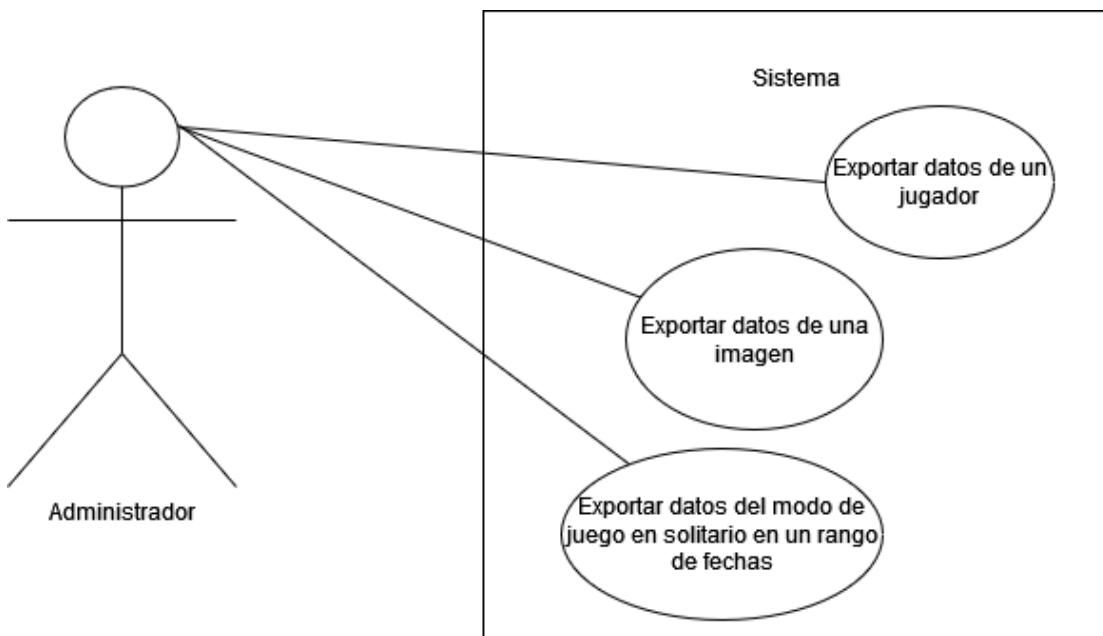


Figura 4.12: Diagrama de caso de uso exportar datos

Especificación de actores del sistema

ACT-0001	Usuario no registrado
Versión	1.0 (15/10/2023)
Descripción	Este actor representa a uno de los usuarios del sistema que puede jugar partidas de forma local y potencialmente puede llegar a registrarse.
Comentarios	Mientras no esté registrado el sistema no guardará datos suyos, por lo que las partidas que juegue no contarán para sus estadísticas.

Cuadro 4.14: Actor usuario no registrado

ACT-0002	Usuario registrado
Versión	1.0 (15/10/2023)
Descripción	Este actor representa a uno de los usuarios del sistema que está registrado.
Comentarios	Al estar registrado, puede acceder a partidas en línea y se recogen sus estadísticas.

Cuadro 4.15: Actor usuario registrado

 ACT-0003	Usuario registrado 2
Versión	1.0 (15/10/2023)
Descripción	Este actor representa a uno de los usuarios del sistema que está registrado.
Comentarios	Ninguno

Cuadro 4.16: Actor usuario no registrado 2

 ACT-0004	Administrador
Versión	1.0 (18/11/2023)
Descripción	Este actor representa a los usuarios del sistema que tendrán privilegios para gestionar a los usuarios y a las imágenes.
Comentarios	Ninguno

Cuadro 4.17: Actor administrador

Especificación de casos de uso del sistema

 UC-0001	Jugar una partida en solitario	
Versión	1.0 (15/10/2023)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario quiera jugar una partida en solitario.	
Precondición	PD	
Secuencia normal	Paso	Acción
	1	El actor Usuario no registrado (ACT-0001) decide comenzar una partida en solitario y pulsa sobre el botón para comenzarla
	2	El sistema crea la partida y selecciona un escudo
	3	El actor Usuario no registrado (ACT-0001) intenta adivinar el escudo hasta que lo consigue o agota sus intentos
	4	El sistema determina el resultado de la partida y lo almacena
Postcondición	El resultado de la partida se almacena correctamente.	
Comentarios	Ninguno	

Cuadro 4.18: Caso de uso jugar una partida en solitario

<input checked="" type="radio"/> UC-0002	Jugar una partida uno contra uno en local	
Versión	1.0 (15/10/2023)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario y alguien más decidan jugar una partida enfrentándose en el mismo dispositivo.	
Precondición	Ambos jugadores están juntos, por lo que podrán utilizar el mismo dispositivo.	
Secuencia normal	Paso	Acción
	1	El actor Usuario no registrado (ACT-0001) y alguien más, deciden enfrentarse en una partida utilizando un solo dispositivo, por lo que introducen sus nombres en el sistema
	2	El sistema inicializa la partida, selecciona un escudo para cada jugador y comienza el turno del primer jugador
	3	El actor Usuario no registrado (ACT-0001) que actúa como jugador número uno, intenta adivinar el escudo hasta que lo consigue o hasta que gasta sus intentos
	4	El sistema determina si el jugador ha acertado y comienza el turno del segundo jugador
	5	El actor Usuario no registrado (ACT-0001) que actúa como jugador número dos, intenta adivinar su escudo hasta que lo consigue o hasta gastar sus intentos
	6	El sistema determina si el jugador ha acertado y a partir de las estadísticas de ambos, decide quien es el ganador y almacena el resultado
Postcondición	El resultado de la partida se almacena correctamente.	
Comentarios	Ninguno	

Cuadro 4.19: Caso de uso jugar una partida uno contra uno en local

UC-0003	Crear una partida en línea	
Versión	1.0 (15/10/2023)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un jugador quiere crear una partida en línea o durante la realización de los siguientes casos de uso: [UC-0005] Jugar una partida en línea.	
Precondición	Un jugador quiere crear una partida en línea.	
Secuencia normal	Paso	Acción
	1	El actor Usuario registrado (ACT-0002) solicita a la aplicación crear una partida en línea
	2	El sistema crea la partida y permite que otro usuario registrado pueda unirse
Postcondición	La partida en línea se crea correctamente.	
Comentarios	Ninguno	

Cuadro 4.20: Caso de uso crear una partida en línea

UC-0004	Unirse a una partida en línea	
Versión	1.0 (15/10/2023)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un jugador quiera unirse a una partida en línea o durante la realización de los siguientes casos de uso: [UC-0005] Jugar una partida en línea.	
Precondición	Debe haber alguna partida en línea ya creada.	
Secuencia normal	Paso	Acción
	1	El actor Usuario registrado 2 (ACT-0003) solicita unirse a una partida en línea
	2	El sistema añade al jugador a una de las partidas disponibles de forma aleatoria
Postcondición	El jugador se une a una partida aleatoria.	
Comentarios	Ninguno	

Cuadro 4.21: Caso de uso unirse a una partida en línea

UC-0005	Jugar una partida en línea	
Versión	1.0 (15/10/2023)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando dos jugadores quieran jugar una partida en línea.	
Precondición	Dos jugadores están ya unidos a una partida en línea.	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Crear una partida en línea (UC-0003)
	2	Se realiza el caso de uso Unirse a una partida en línea (UC-0004)
	3	El sistema inicializa la partida cuando ambos jugadores están listos
	4	El actor Usuario registrado (ACT-0002) intenta adivinar sus tres escudos hasta que lo consigue o hasta que gasta todos sus intentos
	5	El actor Usuario registrado 2 (ACT-0003) intenta adivinar sus tres escudos hasta que lo consigue o hasta que gasta todos sus intentos
	6	El sistema determina el ganador cuando ambos han terminado y almacena el resultado
Postcondición	El resultado de la partida se almacena correctamente.	
Comentarios	Ninguno	

Cuadro 4.22: Caso de uso jugar una partida en línea

UC-0006	Registro de usuarios	
Versión	1.0 (16/10/2023)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario que no está registrado quiera registrarse.	
Precondición	El usuario no está registrado en el sistema.	
Secuencia normal	Paso	Acción
	1	El actor Usuario no registrado (ACT-0001) decide registrarse en el sistema
	2	El sistema solicita al usuario sus datos
	3	El actor Usuario no registrado (ACT-0001) rellena el formulario con sus datos
	4	El sistema guarda los datos del usuario en el sistema
	5	
Postcondición	El sistema guarda los datos del usuario en el sistema.	
Comentarios	Ninguno	

Cuadro 4.23: Caso de uso registro de usuarios

 UC-0007	Inicio de sesión	
Versión	1.0 (16/10/2023)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario registrado quiera iniciar sesión o durante la realización de los siguientes casos de uso: [UC-0008] Consultar mi perfil , [UC-0010] Consultar mis estadísticas .	
Precondición	El usuario debe estar registrado en el sistema.	
Secuencia normal	Paso	Acción
	1	El actor Usuario registrado (ACT-0002) solicita iniciar sesión al sistema
	2	El sistema muestra el formulario de inicio de sesión
	3	El actor Usuario registrado (ACT-0002) rellena el formulario con sus credenciales
	4	Si las credenciales son correctas, el sistema cambia el estado del jugador a identificado
	5	Si las credenciales son erroneas, el sistema comunica al usuario que las credenciales eran incorrectas
Postcondición	El usuario habrá iniciado sesión.	
Comentarios	Ninguno	

Cuadro 4.24: Caso de uso inicio de sesión

 UC-0008	Consultar mi perfil	
Versión	1.0 (25/01/2024)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario identificado quiera ver su perfil o durante la realización de los siguientes casos de uso: [UC-0010] Consultar mis estadísticas , [UC-0014] Edición del perfil , [UC-0015] Cambio de contraseña .	
Precondición	El usuario debe estar registrado.	
Secuencia normal	Paso	Acción
	1	Si el usuario no ha iniciado sesión, se realiza el caso de uso Inicio de sesión (UC-0007)
	2	El actor Usuario registrado (ACT-0002) solicita al sistema ver su perfil
	3	El sistema muestra al usuario su perfil
Postcondición	PD	
Comentarios	Ninguno	

Cuadro 4.25: Caso de uso consultar mi perfil

UC-0009	Consultar estadísticas generales	
Versión	1.0 (25/01/2024)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario quiera consultar las estadísticas de los modos de juego en solitario y en línea o durante la realización de los siguientes casos de uso: [UC-0010] Consultar mis estadísticas , [UC-0025] Exportar datos de partidas en solitario .	
Precondición	PD	
Secuencia normal	Paso	Acción
	1	El actor Usuario no registrado (ACT-0001) solicita al sistema las estadísticas generales
	2	El sistema muestra al usuario las estadísticas de los modos en solitario y en línea
Postcondición	PD	
Comentarios	Ninguno	

Cuadro 4.26: Caso de uso consultar estadísticas generales

UC-0010	Consultar mis estadísticas	
Versión	1.0 (25/01/2024)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario identificado quiera ver sus propias estadísticas.	
Precondición	El usuario debe estar registrado.	
Secuencia normal	Paso	Acción
	1	Si el usuario no ha iniciado sesión, se realiza el caso de uso Inicio de sesión (UC-0007)
	2	Si el usuario solo quiere ver sus estadísticas, se realiza el caso de uso Consultar mi perfil (UC-0008)
	3	El sistema una vez en el perfil del jugador, mostrará las estadísticas del mismo
	4	Si quiere ver las estadísticas generales además de las suyas propias, se realiza el caso de uso Consultar estadísticas generales (UC-0009)
	5	El sistema le mostrará sus estadísticas además de las generales ya que ha iniciado sesión
	6	Si el usuario tiene el rol de administrador, el sistema le permitirá ver estadísticas de las imágenes en las partidas en solitario
Postcondición	PD	
Comentarios	Ninguno	

Cuadro 4.27: Caso de uso consultar mis estadísticas

<input type="radio"/> UC-0011	Ver listado de imágenes	
Versión	1.0 (25/01/2024)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador quiera acceder al listado de imágenes del sistema o durante la realización de los siguientes casos de uso: [UC-0012] Consultar datos de una imagen , [UC-0017] Deshabilitar una imagen , [UC-0018] Habilitar una imagen .	
Precondición	El usuario debe tener el rol de administrador.	
Secuencia normal	Paso	Acción
	1	El actor Administrador (ACT-0004) solicita al sistema el listado de imágenes
	2	El sistema muestra al administrador un listado paginado con todas las imágenes del sistema
	Postcondición	
Comentarios	Ninguno	

Cuadro 4.28: Caso de uso ver listado de imágenes

<input type="radio"/> UC-0012	Consultar datos de una imagen	
Versión	1.0 (27/01/2024)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador quiera consultar los datos de una imagen o durante la realización de los siguientes casos de uso: [UC-0024] Exportar datos de una imagen .	
Precondición	El usuario debe tener el rol de administrador.	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Ver listado de imágenes (UC-0011)
	2	El actor Administrador (ACT-0004) selecciona la imagen de la cual quiera consultar los datos
	3	El sistema muestra al administrador los datos de la imagen además de unas estadísticas de la misma en las partidas en solitario
Postcondición	PD	
Comentarios	Ninguno	

Cuadro 4.29: Caso de uso consultar datos de una imagen

 UC-0013	Consultar el manual	
Versión	1.0 (27/01/2024)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario quiera consultar el manual.	
Precondición	PD	
Secuencia normal	Paso	Acción
	1	El actor Usuario no registrado (ACT-0001) solicita al sistema ver el manual
	2	El sistema muestra el manual al usuario
	3	Si el usuario tiene permisos de administrador, el sistema le mostrará también el manual del administrador
Postcondición	PD	
Comentarios	Ninguno	

Cuadro 4.30: Caso de uso consultar el manual

 UC-0014	Edición del perfil	
Versión	1.0 (27/01/2024)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario quiera editar sus datos.	
Precondición	El usuario debe haber iniciado sesión.	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Consultar mi perfil (UC-0008)
	2	El actor Usuario registrado (ACT-0002) pulsará sobre el botón de edición del perfil
	3	El sistema mostrará el formulario de edición de perfil relleno con los datos actuales del jugador
	4	El actor Usuario registrado (ACT-0002) modificará los campos que desee y enviará el formulario
	5	El sistema confirmará si los datos son válidos y modificará los datos del usuario
Postcondición	Los datos del usuario serán modificados correctamente.	
Comentarios	Ninguno	

Cuadro 4.31: Caso de uso edición del perfil

 UC-0015	Cambio de contraseña	
Versión	1.0 (27/01/2024)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un usuario quiera cambiar su contraseña.	
Precondición	El usuario debe haber iniciado sesión.	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Consultar mi perfil (UC-0008)
	2	El actor Usuario registrado (ACT-0002) pulsará sobre el botón de cambio de contraseña
	3	El sistema mostrará el formulario de cambio de contraseña
	4	El actor Usuario registrado (ACT-0002) llenará el formulario con su antigua contraseña y la nueva
	5	El sistema actualizará la contraseña del usuario y la guardará encriptada
Postcondición	La contraseña del usuario se actualizará y se guardará de forma segura.	
Comentarios	Ninguno	

Cuadro 4.32: Caso de uso cambio de contraseña

 UC-0016	Añadir una nueva imagen	
Versión	1.0 (27/01/2024)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador quiera añadir una nueva imagen al sistema.	
Precondición	La imagen no debe estar registrada en el sistema.	
Secuencia normal	Paso	Acción
	1	El actor Administrador (ACT-0004) solicita añadir una nueva imagen
	2	El sistema muestra el formulario
	3	El actor Administrador (ACT-0004) rellena el formulario y sube el archivo de la imagen
	4	El sistema registra la nueva imagen y esta podrá aparecer en las partidas a partir de este momento
Postcondición	La imagen quedará registrada en el sistema.	
Comentarios	Ninguno	

Cuadro 4.33: Caso de uso añadir una nueva imagen

 UC-0017	Deshabilitar una imagen	
Versión	1.0 (27/01/2024)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador quiera deshabilitar una imagen para que no aparezca en las partidas.	
Precondición	La imagen debe existir en el sistema y tiene que estar habilitada.	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Ver listado de imágenes (UC-0011)
	2	El actor Administrador (ACT-0004) pulsa sobre el botón para deshabilitar la imagen que el quiera
	3	El sistema deshabilita la imagen
Postcondición	La imagen quedará deshabilitada y dejará de aparecer en las partidas.	
Comentarios	Ninguno	

Cuadro 4.34: Caso de uso deshabilitar una imagen

 UC-0018	Habilitar una imagen	
Versión	1.0 (27/01/2024)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador quiera habilitar una imagen para que no aparezca en las partidas.	
Precondición	La imagen debe existir en el sistema y tiene que estar deshabilitada.	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Ver listado de imágenes (UC-0011)
	2	El actor Administrador (ACT-0004) pulsa sobre el botón para habilitar la imagen que el quiera
	3	El sistema habilita la imagen
Postcondición	La imagen quedará habilitada y podrá volver a aparecer en las partidas.	
Comentarios	Ninguno	

Cuadro 4.35: Caso de uso habilitar una imagen

 UC-0019	Ver listado de jugadores	
Versión	1.0 (27/01/2024)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando el administrador quiera acceder al listado de jugadores del sistema o durante la realización de los siguientes casos de uso: [UC-0020] Consultar datos de un jugador , [UC-0021] Deshabilitar un jugador , [UC-0022] Habilitar un jugador .	
Precondición	El usuario debe tener el rol de administrador.	
Secuencia normal	Paso	Acción
	1	El actor Administrador (ACT-0004) solicita al sistema el listado de jugadores
	2	El sistema muestra al administrador un listado paginado con todos los jugadores del sistema
Postcondición	PD	
Comentarios	Ninguno	

Cuadro 4.36: Caso de uso ver listado de jugadores

 UC-0020	Consultar datos de un jugador	
Versión	1.0 (27/01/2024)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador quiera consultar los datos de una imagen o durante la realización de los siguientes casos de uso: [UC-0023] Exportar datos de un jugador .	
Precondición	El usuario debe tener permisos de administrador	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Ver listado de jugadores (UC-0019)
	2	El actor Administrador (ACT-0004) selecciona el jugador del cual quiera consultar los datos
	3	El sistema muestra al administrador los datos del jugador además de las estadísticas del mismo
Postcondición	PD	
Comentarios	Ninguno	

Cuadro 4.37: Caso de uso consultar datos de un jugador

<input type="radio"/> UC-0021	Deshabilitar un jugador	
Versión	1.0 (27/01/2024)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador quiera deshabilitar un jugador para que no pueda volver a iniciar sesión.	
Precondición	El jugador debe estar registrado en el sistema y tiene que estar habilitado.	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Ver listado de jugadores (UC-0019)
	2	El actor Administrador (ACT-0004) pulsa sobre el botón para deshabilitar al jugador que el quiera.
	3	El sistema deshabilita al jugador
Postcondición	El jugador quedará deshabilitado y no podrá volver a iniciar sesión.	
Comentarios	Ninguno	

Cuadro 4.38: Caso de uso deshabilitar un jugador

<input type="radio"/> UC-0022	Habilitar un jugador	
Versión	1.0 (27/01/2024)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador quiera habilitar a un jugador para que pueda volver a iniciar sesión.	
Precondición	El jugador debe estar registrado en el sistema y debe estar deshabilitado.	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Ver listado de jugadores (UC-0019)
	2	El actor Administrador (ACT-0004) pulsa sobre el botón para habilitar al jugador que el quiera
	3	El sistema habilita de nuevo al jugador
Postcondición	El jugador quedará habilitado y podrá volver a iniciar sesión.	
Comentarios	Ninguno	

Cuadro 4.39: Caso de uso habilitar un jugador

 UC-0023	Exportar datos de un jugador	
Versión	1.0 (27/01/2024)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador quiera exportar las estadísticas de las partidas en solitario de un jugador.	
Precondición	El jugador debe estar registrado en el sistema.	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Consultar datos de un jugador (UC-0020)
	2	El actor Administrador (ACT-0004) pulsa el botón de exportar estadísticas
	3	El sistema genera un archivo json con las estadísticas del jugador y hace que el navegador del usuario lo descargue
Postcondición	El administrador recibirá un archivo json con las estadísticas del jugador.	
Comentarios	Ninguno	

Cuadro 4.40: Caso de uso exportar datos de un jugador

 UC-0024	Exportar datos de una imagen	
Versión	1.0 (27/01/2024)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador quiera exportar las estadísticas de las partidas en solitario en las que ha aparecido una imagen.	
Precondición	La imagen debe estar registrada en el sistema.	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Consultar datos de una imagen (UC-0012)
	2	El actor Administrador (ACT-0004) pulsa el botón de exportar estadísticas
	3	El sistema genera un archivo json con las estadísticas de la imagen y hace que el navegador del usuario lo descargue
Postcondición	El administrador recibirá un archivo json con las estadísticas de la imagen.	
Comentarios	Ninguno	

Cuadro 4.41: Caso de uso exportar datos de una imagen

UC-0025	Exportar datos de partidas en solitario	
Versión	1.0 (27/01/2024)	
Descripción	El sistema deberá comportarse tal como se describe en el siguiente caso de uso cuando un administrador quiera exportar los datos de las partidas en solitario en un rango de fechas.	
Precondición	PD	
Secuencia normal	Paso	Acción
	1	Se realiza el caso de uso Consultar estadísticas generales (UC-0009)
	2	El actor Administrador (ACT-0004) llenará el formulario con el rango de fechas del que quiere las partidas
	3	El sistema devolverá los datos de todas las partidas en solitario que se han jugado en el rango de fechas en un archivo json que será descargado por el navegador del usuario
Postcondición	El administrador recibirá un archivo json con las estadísticas de la imagen.	
Comentarios	Ninguno	

Cuadro 4.42: Caso de uso exportar datos de partidas en solitario

Requisitos funcionales

Requisitos de información del sistema

IRQ-0001	Información sobre los usuarios
Versión	1.0 (16/10/2023)
Descripción	El sistema deberá almacenar la información correspondiente a los usuarios. En concreto:
Datos específicos	<ul style="list-style-type: none"> • Nombre de usuario • Contraseña • Correo electrónico • Nombre
Comentarios	Ninguno

Cuadro 4.43: Requisito de información sobre los usuarios

 IRQ-0002	Información sobre las partidas en solitario
Versión	1.0 (16/10/2023)
Descripción	El sistema deberá almacenar la información correspondiente a las partidas en solitario. En concreto:
Datos específicos	<ul style="list-style-type: none"> • Escudo seleccionado • Usuario (si está identificado) • Número de intentos • Resultado de la partida • Fecha de la partida
Comentarios	Ninguno

Cuadro 4.44: Requisito de información sobre las partidas en solitario

 IRQ-0003	Información sobre las partidas multijugador en línea
Versión	1.0 (16/10/2023)
Descripción	El sistema deberá almacenar la información correspondiente a las partidas multijugador en línea. En concreto:
Datos específicos	<ul style="list-style-type: none"> • Escudos seleccionados para el jugador 1 • Escudos seleccionados para el jugador 2 • Usuario 1 • Usuario 2 • Intentos jugador 1 • Intentos jugador 2 • Aciertos jugador 1 • Aciertos jugador 2 • Ganador • Duración jugador 1 • Duración jugador 2
Comentarios	Ninguno

Cuadro 4.45: Requisito de información sobre las partidas multijugador en línea

 IRQ-0004	Información sobre las partidas multijugador locales
Versión	1.0 (16/10/2023)
Descripción	El sistema deberá almacenar la información correspondiente a las partidas multijugador locales. En concreto:
Datos específicos	<ul style="list-style-type: none"> • Nombre jugador 1 • Nombre jugador 2 • Imagen seleccionada jugador 1 • Imagen seleccionada jugador 2 • Intentos jugador 1 • Intentos jugador 2 • ¿Ha acertado el jugador 1? • ¿Ha acertado el jugador 2? • Ganador • Duración jugador 1 • Duración jugador 2
Comentarios	Ninguno

Cuadro 4.46: Requisito de información sobre las partidas multijugador locales

 IRQ-0005	Información sobre las imágenes
Versión	1.0 (18/11/2023)
Descripción	El sistema deberá almacenar la información correspondiente a las imágenes. En concreto:
Datos específicos	<ul style="list-style-type: none"> • Nombre completo de la imagen • Localización de la imagen • ¿Tiene año? • ¿Tiene iniciales? • ¿Tiene nombre?
Comentarios	Ninguno

Cuadro 4.47: Requisito de información sobre las imágenes

Requisitos de reglas de negocio del sistema

<input checked="" type="checkbox"/> CRQ-0001	Restricción sobre el tipo de partida
Versión	1.0 (16/10/2023)
Descripción	El sistema deberá respetar la siguiente regla de negocio o restricción: Los usuarios no registrados no deben poder acceder a partidas en línea.
Comentarios	Ninguno

Cuadro 4.48: Regla de negocio sobre el tipo de partida

<input checked="" type="checkbox"/> CRQ-0002	Restricción sobre las estadísticas
Versión	1.0 (16/10/2023)
Descripción	El sistema deberá respetar la siguiente regla de negocio o restricción: Los usuarios no identificados solo podrán ver las estadísticas generales.
Comentarios	Ninguno

Cuadro 4.49: Regla de negocio sobre las estadísticas

<input checked="" type="checkbox"/> CRQ-0003	Restricción sobre partidas en solitario
Versión	1.0 (27/01/2024)
Descripción	El sistema deberá respetar la siguiente regla de negocio o restricción: Una partida en solitario solo debe ser accesible por el jugador que la crea, aunque este no esté registrado.
Comentarios	Ninguno

Cuadro 4.50: Regla de negocio sobre las partidas en solitario

<input checked="" type="checkbox"/> CRQ-0004	Restricción sobre partidas multijugador locales
Versión	1.0 (27/01/2024)
Descripción	El sistema deberá respetar la siguiente regla de negocio o restricción: Una partida uno contra uno en local solo debe ser accesible desde el dispositivo en el que se crea la partida.
Comentarios	Ninguno

Cuadro 4.51: Regla de negocio sobre las partidas multijugador locales

<input checked="" type="checkbox"/> CRQ-0005	Restricción sobre partidas en línea
Versión	1.0 (27/01/2024)
Descripción	El sistema deberá respetar la siguiente regla de negocio o restricción: Una partida en línea solo debe ser accesible para el jugador que crea la partida y para el que se une.
Comentarios	Ninguno

Cuadro 4.52: Regla de negocio sobre las partidas en línea

<input checked="" type="checkbox"/> CRQ-0006	Restricción sobre el abandono de partidas
Versión	1.0 (27/01/2024)
Descripción	El sistema deberá respetar la siguiente regla de negocio o restricción: Si un jugador abandona una partida en línea, automáticamente, el otro jugador se convertirá en ganador de la partida.
Comentarios	Ninguno

Cuadro 4.53: Regla de negocio sobre el abandono de partidas

<input checked="" type="checkbox"/> CRQ-0007	Restricción contra la pasividad en partidas en línea
Versión	1.0 (27/01/2024)
Descripción	El sistema deberá respetar la siguiente regla de negocio o restricción: Si un jugador tarda más de siete minutos en jugar una partida en línea, esta se le dará por abandonada.
Comentarios	Ninguno

Cuadro 4.54: Regla de negocio contra la pasividad en partidas en línea

<input checked="" type="checkbox"/> CRQ-0008	Restricción sobre partidas no comenzadas
Versión	1.0 (27/01/2024)
Descripción	El sistema deberá respetar la siguiente regla de negocio o restricción: Si una partida lleva creada más de cinco minutos y aun no ha comenzado, será eliminada para evitar que uno de los dos jugadores no esté activo y por eso nunca comience la partida.
Comentarios	Ninguno

Cuadro 4.55: Regla de negocio sobre las partidas no comenzadas

Requisitos no funcionales

Requisitos de seguridad del sistema

 NFR-0001	Protección de usuarios
Versión	1.0 (16/10/2023)
Descripción	El sistema deberá encriptar las contraseñas de los usuarios antes de almacenarlas.
Comentarios	Ninguno

Cuadro 4.56: Requisito de seguridad sobre la protección de usuarios

 NFR-0002	Protección SQLInjection
Versión	1.0 (16/10/2023)
Descripción	El sistema deberá estar protegido frente a inyecciones de SQL.
Comentarios	Ninguno

Cuadro 4.57: Requisito de seguridad protección SQLInjection

 NFR-0003	Protección ScriptInjection
Versión	1.0 (16/10/2023)
Descripción	El sistema deberá estar protegido frente a inyecciones de Scripts.
Comentarios	Ninguno

Cuadro 4.58: Requisito de seguridad protección ScriptInjection

 NFR-0004	Protección CSRF
Versión	1.0 (16/10/2023)
Descripción	El sistema deberá estar protegido frente a ataques CSRF.
Comentarios	Ninguno

Cuadro 4.59: Requisito de seguridad protección CSRF

Información sobre trazabilidad

 TRM-0002 A- A+	OBJ-0001	OBJ-0002	OBJ-0003	OBJ-0004	OBJ-0005	OBJ-0006
IRQ-0001	-	-	-	▲	-	-
IRQ-0002	▲	-	-	-	-	-
IRQ-0003	-	-	-	-	-	-
IRQ-0004	-	▲	▲	-	-	-
IRQ-0005	-	-	-	-	▲	-

Cuadro 4.60: Matriz de trazabilidad de los requisitos de información

 TRM-0003	OBJ-0001	OBJ-0002	OBJ-0003	OBJ-0004	OBJ-0005	OBJ-0006
A- A+						
UC-0001	⬆	-	-	-	-	-
UC-0002	-	⬆	-	-	-	-
UC-0003	-	-	⬆	-	-	-
UC-0004	-	-	⬆	-	-	-
UC-0005	-	-	⬆	-	-	-
UC-0006	-	-	-	⬆	-	-
UC-0007	-	-	-	⬆	-	-
UC-0008	-	-	-	⬆	-	-
UC-0009	-	-	-	⬆	-	-
UC-0010	-	-	-	⬆	-	-
UC-0011	-	-	-	-	⬆	-
UC-0012	-	-	-	-	⬆	-
UC-0013	-	-	-	⬆	-	-
UC-0014	-	-	-	⬆	-	-
UC-0015	-	-	-	⬆	-	-
UC-0016	-	-	-	-	⬆	-
UC-0017	-	-	-	-	⬆	-
UC-0018	-	-	-	-	⬆	-
UC-0019	-	-	-	-	⬆	-
UC-0020	-	-	-	-	⬆	-
UC-0021	-	-	-	-	⬆	-
UC-0022	-	-	-	-	⬆	-
UC-0023	-	-	-	-	-	⬆
UC-0024	-	-	-	-	-	⬆
UC-0025	-	-	-	-	-	⬆

Cuadro 4.61: Matriz de trazabilidad sobre los casos de uso

 TRM-0004	OBJ-0001	OBJ-0002	OBJ-0003	OBJ-0004	OBJ-0005	OBJ-0006
	A-	A+				
CRQ-0001	⬆	⬆	⬆	⬆	-	-
CRQ-0002	-	-	-	⬆	-	-
CRQ-0003	⬆	-	-	-	-	-
CRQ-0004	-	⬆	-	-	-	-
CRQ-0005	-	-	⬆	-	-	-
CRQ-0006	-	-	⬆	-	-	-
CRQ-0007	-	-	⬆	-	-	-
CRQ-0008	-	-	⬆	-	-	-

Cuadro 4.62: Matriz de trazabilidad sobre las reglas de negocio

 TRM-0005	OBJ-0001	OBJ-0002	OBJ-0003	OBJ-0004	OBJ-0005	OBJ-0006
	A-	A+				
NFR-0001	-	-	-	⬆	-	-
NFR-0002	-	-	-	⬆	-	-
NFR-0003	-	-	-	⬆	-	-
NFR-0004	-	-	-	-	⬆	-

Cuadro 4.63: Matriz de trazabilidad sobre los requisitos no funcionales

Diagrama de la arquitectura del sistema

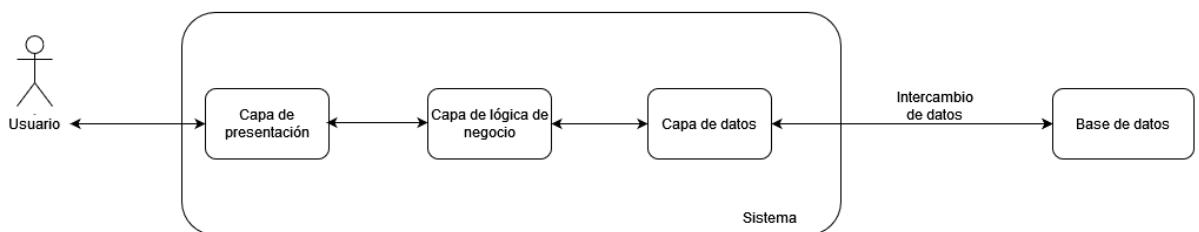


Figura 4.13: Diagrama de la arquitectura del sistema

La arquitectura pensada para el sistema se basa en una arquitectura por capas donde todos los usuarios se pueden conectar al sistema interactuando con una capa y cada capa proveerá de sus servicios a la capa inmediatamente superior para que la aplicación funcione correctamente. Las capas definidas para el sistema serán las siguientes:

- Una capa de presentación que será la capa con la que los usuarios interactuarán y dependiendo del usuario podrá acceder a unas funcionalidades u otras, por ejemplo, los usuarios no registrados no podrán ver sus propias estadísticas ni podrán acceder a partidas en línea y las funciones del módulo de administración tan solo serán accesibles para aquellos usuarios que tengan el rol de administradores.
- Una capa de lógica de negocio que se encargará de hacer los cálculos necesarios para poder proveer de funcionalidad a la aplicación.
- Una capa de datos que será la encargada de interactuar con la base de datos tanto para introducir como para modificar como para solicitar datos, esta capa será la encargada de proporcionarle a la capa de lógica los datos que esta necesite para realizar sus cálculos.

Modelo de clases del sistema

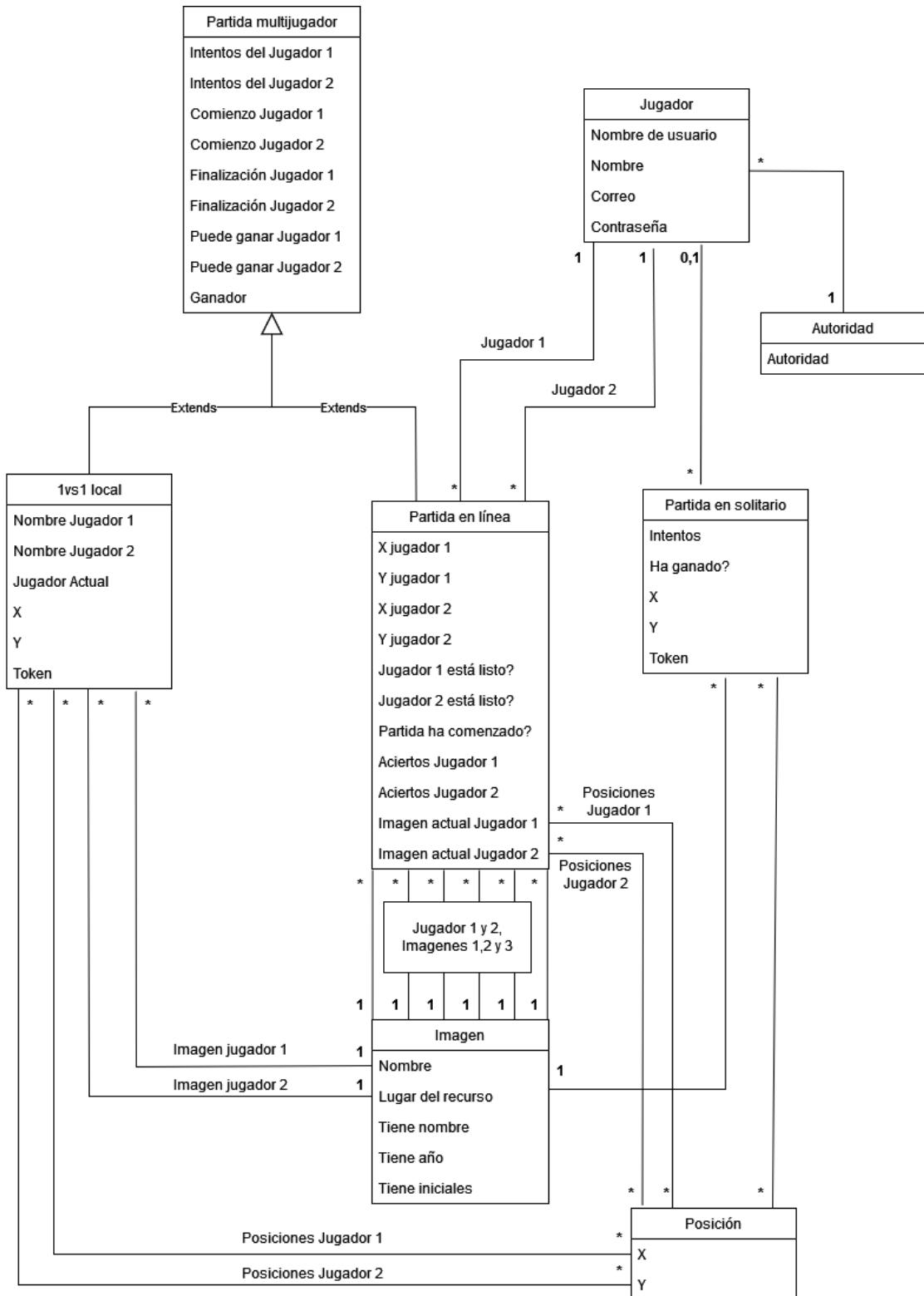


Figura 4.14: Modelo de clases del sistema

Descripción de las clases del sistema

- **Jugador:** Clase que representa a cualquier persona que use el sistema, dependiendo de si el usuario se ha identificado o no, podrá acceder a unas funciones de la aplicación o a otras.
- **Autoridad:** Representa la autoridad que tiene un usuario, a partir de la autoridad se determina si puede acceder a unas funciones u otras, por ejemplo solo quienes tengan la autoridad de administrador podrán acceder a las funciones de administrador.
- **Posición:** Entidad que representa una porción de una imagen, esta entidad es necesaria para asegurarnos que a lo largo de una partida no se muestra dos veces la misma porción de una imagen como pista.
- **Imagen:** Representa cualquier escudo almacenado en la base de datos.
- **Partida en solitario:** Clase utilizada para representar las partidas en solitario, almacenando el escudo seleccionado, los intentos, el resultado, las porciones ya mostradas y la información de la porción actual. Al no ser obligatorio estar registrado para jugar, la relación con la clase Jugador es opcional.
- **Partida multijugador:** Representa cualquier partida en la que estén involucrados más de un jugador, ya sea de forma local o en línea, por eso engloba datos correspondientes a la participación de ambos jugadores en la partida.
 - **Partida 1vs1 local:** Clase que representa las partidas multijugador de forma local, como para jugarlas no hace falta estar identificado y no pueden identificarse dos usuarios distintos al mismo tiempo en el mismo dispositivo, no tienen ninguna relación con la clase Jugador.
 - **Partida en línea :** Clase que representa las partidas multijugador en línea, es obligatorio estar identificado para jugar, por eso hay una relación obligatoria con la clase Jugador.

Diseño

Diagrama de dominio/diseño

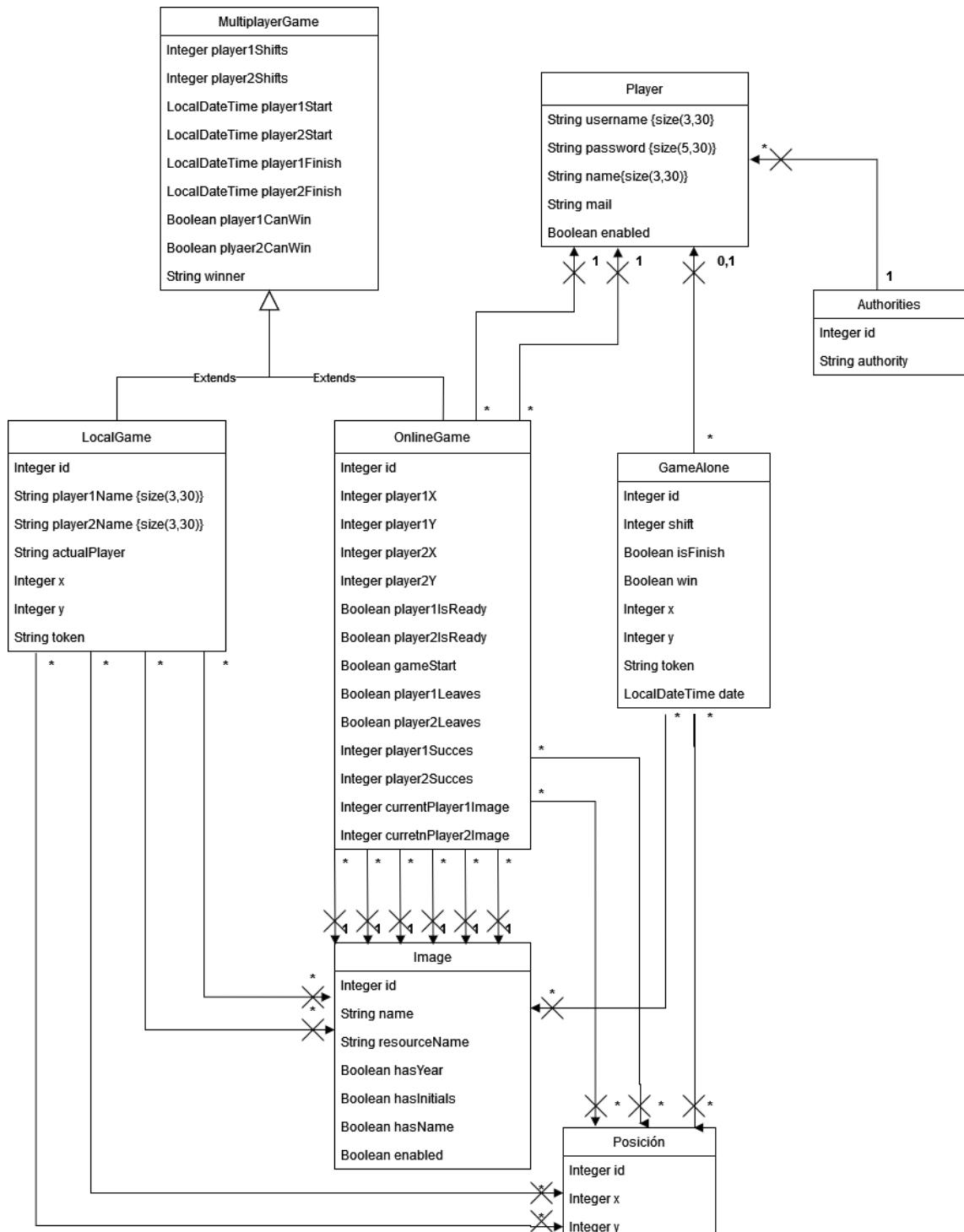


Figura 4.15: Diagrama dominio/diseño del sistema

Diagrama de capas

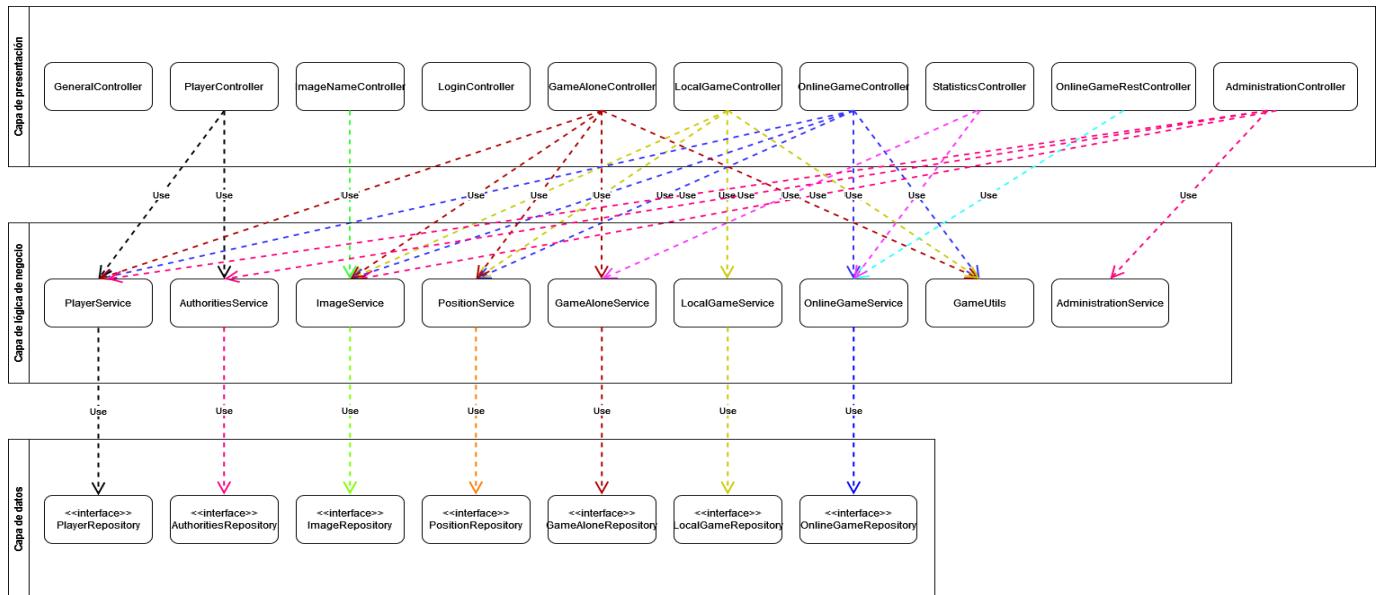


Figura 4.16: Diagrama de capas del sistema

Patrones de diseño y arquitectónicos aplicados

En esta sección se especificará el conjunto de patrones de diseño y arquitectónicos aplicados durante el proyecto.

Patrón Modelo-vista-controlador¹

La aplicación completa está diseñada con el patrón MVC por el que estructuramos la aplicación en tres componentes distintos:

- **Modelo:** Es la representación de la información en la que incluiremos los datos y la lógica de negocio necesaria para trabajar con estos datos.
- **Vista:** Es la representación del modelo en la manera que le usuario pueda interactuar con este. Es nuestra interfaz de usuario.
- **Controlador:** Responde a los eventos de la interfaz de usuario que se realizan mediante peticiones HTTP en las que invocamos los modelos y, probablemente, las propias vistas.

Se ha decidido utilizar este patrón ya que facilita la separación de las preocupaciones, lo que permite que el código sea más modular y fácil de mantener. Además, de dar la posibilidad de reutilizar componentes como modelos o vistas.

Patrón Front Controllerⁱⁱ

Este patrón ha sido utilizado en todos los controladores ya que con esta función controlamos todas las peticiones HTTP de la aplicación. Esto se realiza gracias al ‘Dispatcher Servlet’ que provee Spring, que invoca el método de controlador correspondiente a cada petición.

Una de las ventajas que nos proporciona este patrón es que nos permite controlar todas las peticiones que recibe la página web de forma automática. Por otro lado, podremos añadir nuevos controladores cuando queremos sin afectar el funcionamiento de los ya existentes. Además, el ‘Dispatcher Servlet’ permite centrarse en la lógica de negocio y la presentación de las vistas sin tener que preocuparse por los detalles de la gestión de solicitudes HTTP.

Patrón de repositoriosⁱⁱⁱ

El patrón repositorio de datos es un patrón arquitectónico que se utiliza para separar la lógica que recupera los datos de la base de datos de la lógica de negocio. Los repositorios nos proporcionan una capa de abstracción entre la aplicación y los datos, lo que hace que el acceso y la manipulación de estos, sea más sencillo y eficiente.

Algunas de las ventajas que obtenemos utilizando este patrón, son, la abstracción de la base de datos, ya que el repositorio oculta los detalles de cómo se almacenan y recuperan los datos. La separación de preocupaciones ya que nos permite separar la lógica de negocio y el acceso a los datos. Además, se facilitan las pruebas unitarias y la reutilización de código.

Patrón Service Layer^{iv}

El patrón Service Layer proporciona una interfaz cohesiva y simplificada a la lógica de negocio de una aplicación, lo que permite una mejor organización, modularidad y mantenimiento del código. Este patrón se basa en agrupar la lógica del negocio en una capa de servicios que interactúa con los controladores y los repositorios.

Al utilizar el patrón Service Layer, aprovechamos ventajas como la separación de preocupaciones, la reutilización del código, la mejora en la organización del código y la facilidad para las pruebas unitarias.

Patrón Dependency Injection^v

El Dependency Injection es un patrón que se utiliza en Spring para administrar las dependencias de los componentes de una aplicación de manera eficiente y flexible. En el patrón Dependency Injection los objetos no se encargan de crear sus propias dependencias, en cambio, las dependencias se inyectan en el objeto desde el exterior, normalmente a través de un contenedor de inversión del control.

Utilizar el patrón Dependency Injection nos aporta desacoplamiento ya que la inyección de dependencias reduce el acoplamiento, lo que facilita la modificación y evolución del código. Además, los componentes se vuelven más reutilizables y las pruebas unitarias se simplifican. Por último, como la gestión de las dependencias se realiza externamente, se nos permite cambiar el comportamiento de la aplicación sin modificar el código fuente.

Patrón Proxy^{vi}

El patrón proxy se utiliza para ayudar al Dependency Injection de forma que el Proxy es un sustituto para cualquier objeto que se solicite. Es decir, el Proxy actúa como una interfaz que controla el acceso al objeto real y permite agregar funcionalidad adicional al acceder a ese objeto.

Se ha utilizado el patrón proxy para lograr mayor eficiencia, seguridad y control de acceso en la creación de objetos costosos.

Patrón Domain Model^{vii}

El patrón Domain Model es un enfoque arquitectónico que se utiliza principalmente en el contexto de sistemas de información y aplicaciones empresariales. El Domain Model se centra en la representación del modelo de datos del dominio, lo que significa que se modelarán las entidades, atributos y relaciones que son relevantes para la aplicación.

En el caso de este proyecto, se ha utilizado el Domain Model ya que es especialmente útil en aplicaciones donde la lógica de negocio es compleja y crítica para el éxito de la aplicación, como “Con Un Pedazo Es Suficiente” es un juego que tiene una lógica de negocio crítica y compleja, al utilizar Domain Model, la lógica del negocio se separa de los detalles de implementación lo que facilita la comprensión de la aplicación y reduce la complejidad de esta misma.

Patrón Identity Field^{viii}

El patrón Identity Field es un patrón utilizado en el contexto del diseño de bases de datos y modelado de datos, particularmente cuando se trabaja con bases de datos relacionales. Este patrón se utiliza para identificar y gestionar la clave primaria o campo de identidad de una tabla, que es un valor único que identifica de forma única a cada elemento de la tabla.

Al utilizar este patrón en la aplicación, se podrá identificar de forma única a cada entidad dentro de la aplicación lo que permitirá acceder rápidamente a los datos de una entidad a partir de su identificador.

Pagination^{ix}

El patrón de diseño paginación es un patrón enfocado a las interfaces de usuarios y se usa cuando tenemos grandes cantidades de datos que mostrar en una lista y solo queremos mostrar parte de ellos para no saturar al usuario.

Las ventajas que obtenemos al aplicar dicho patrón son, mejora en el rendimiento, navegación más sencilla o manejo más eficiente de recursos entre otras.

Decisiones de diseño tomadas

A lo largo del diseño han surgido situaciones en las que para diseñar las distintas funcionalidades de la aplicación han surgido diferentes alternativas, lo que ha conllevado en cada caso un estudio de las diferentes alternativas para acabar tomando la decisión más conveniente para la aplicación.

Decisión de diseño número 1: Almacenamiento de imágenes

Contexto del problema: La aplicación necesitará imágenes para funcionar y la aplicación deberá poder mostrar esas imágenes en las partidas, pero estas imágenes se pueden utilizar de distintas formas.

Alternativa 1: Guardar las imágenes dentro del sistema de archivos de la aplicación, haciendo que se almacenen en la memoria de esta.

Ventajas:

- El tiempo de carga para mostrar las imágenes siempre estaría controlador y se reduciría.
- Podemos controlar tanto el tamaño como el formato de las imágenes.

Desventajas:

- Las imágenes ocuparían espacio en la memoria de la aplicación.

Alternativa 2: Utilizar imágenes de internet y acceder a ellas para mostrarlas a partir de sus URL.

Ventajas:

- Las imágenes no ocuparían espacio en la memoria de la aplicación.

Desventajas:

- Mayor tiempo de carga en cada turno ya que habría que buscar las imágenes en internet para mostrarlas y en condiciones de conexión lenta esto reduciría el rendimiento de la aplicación.

- No controlamos el formato ni el tamaño de las imágenes.

Conclusión:

Tras analizar ambas alternativas la mejor opción era la primera ya que el tiempo de carga de las partidas sería menor y a pesar de que habría que almacenar las imágenes en la memoria de la aplicación, la cantidad de imágenes siempre estaría controlada por mi para que no tuviera efectos adversos en el rendimiento de la aplicación.

Decisión de diseño número 2: Selección de los pedazos

Contexto del problema: A la hora de escoger como se seleccionaban los pedazos que se utilizarían en la aplicación surgieron varias alternativas muy diferentes a nivel conceptual.

Alternativa 1: Escoger un pedazo cualquiera totalmente aleatorio con un tamaño de 500 x 500 pixeles, si el jugador falla, aumentar el tamaño de dicho pedazo a 550 x 550 pixeles para dar un poco más de información, si el jugador vuelve a fallar, aumentar finalmente a 600 x 600 pixeles.



Figura 4.17: Ejemplo primera propuesta para escoger pedazos

Ventajas:

- Al tener un modelo de selección completamente aleatorio, en muchas partidas distintas puede aparecer el mismo escudo y aun así pueden ser pedazos de zonas que nunca nos hayan aparecido en la aplicación.
- Siempre estaríamos jugando con solo un pedazo, como el nombre de la aplicación indica.

Desventajas:

- La complejidad del juego con este modelo es mucho mayor, ya que por partida tan solo mostramos un pedazo de 600 x 600 pixeles,

Alternativa 2: Dividir la imagen en dieciséis pedazos y para las partidas mostrar uno de esos pedazos ya predefinidos y tras cada fallo del usuario desvelar otro pedazo de esos dieciséis.

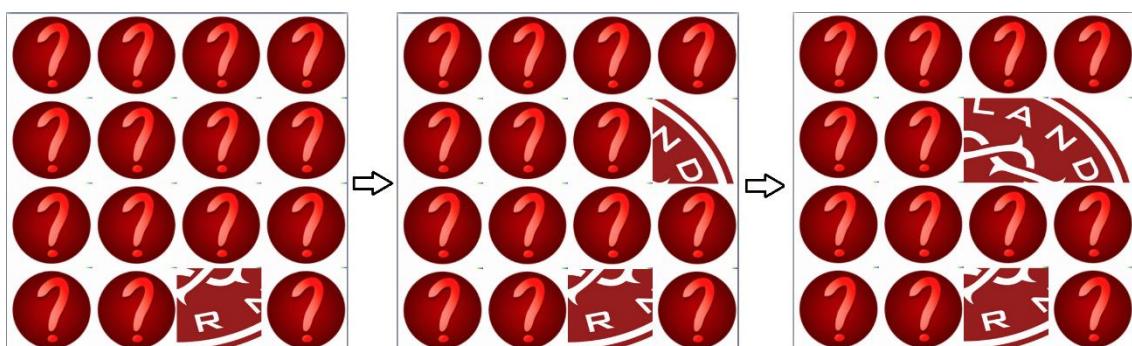


Figura 4.18: Ejemplo segunda propuesta para escoger pedazos

** La imagen de la interrogación que cubre el escudo ha sido generada por la Inteligencia artificial DALL-E 2 <https://labs.openai.com/>

Ventajas:

- La complejidad del juego es menor ya que en cada partida se muestra más de un pedazo en el caso de que usuario no acierte.

Desventajas:

- Como los pedazos no se escogen de forma completamente aleatoria, es decir que siempre se muestra alguno de los dieciséis, al cabo de poco tiempo cualquier usuario habrá visto los dieciséis pedazos de cada escudo.

Conclusión:

Finalmente, tras haberlo consultado con el tutor del TFG, ambos llegamos a la conclusión de que la segunda opción podría ser la mejor al ser un poco más fácil, además de que con la opción de mostrar tan solo un pedazo aleatorio le daríamos muy poca información a los usuarios y esto podría llegar a ser frustrante.

Decisión de diseño número 3: Gestión de los pedazos

Contexto del problema: Una de las funcionalidades del juego donde han surgido problemas ha sido a la hora de mostrar los pedazos de las imágenes en todos los modos de juegos, y es que en cada partida cuando estuviesen determinados los escudos de la partida, habría que ver como mostrábamos los pedazos seleccionados en cada turno.

Alternativa 1: Aprovechar los estilos de la hoja de estilos CSS para mostrar solo una cantidad de pixeles determinado de una imagen.

Ventajas:

- No tiene coste de almacenamiento en memoria y computacionalmente es más rápida ya que la aplicación de estilos CSS no tiene prácticamente coste temporal y son responsabilidad del navegador.

Desventajas:

- No respetaríamos un principio fundamental de la ingeniería del software como lo es la separación de responsabilidades, este principio implica que cada componente debe cumplir única y exclusivamente con sus responsabilidades, y en este caso estaríamos pasándolo por alto, ya que los estilos CSS son estáticos y no cambian, si queremos que cambien, deberíamos irlos generando en algún otro componente como en este caso serían los controladores Java de nuestra aplicación, al estar

generando estilos CSS en nuestros controladores estaríamos violando el principio de separación de responsabilidades. Además, los estilos CSS se suelen aplicar en una hoja de estilos a parte llamada “style.css” y en este caso se estarían aplicando directamente en los jsp.

Alternativa 2: A partir de la imagen original, crear una copia y recortar la copia dejando solo los pedazos que durante el turno se mostrarían al usuario. Por lo que en cada caso habría que generar una imagen nueva y almacenarla durante un tiempo.

Ventajas:

- No incumpliríamos ningún principio fundamental de la ingeniería el software.

Desventajas:

- Generar una nueva imagen con un pedazo de otra requiere un coste temporal ya que la creación de cada imagen se haría píxel a píxel.
- Las imágenes se tendrían que almacenar durante un tiempo en memoria de la aplicación lo que llevaría un gran coste de memoria si hay muchas partidas al mismo tiempo.

Alternativa 3: En la inicialización de los datos de la aplicación, por cada escudo, añadir 17 imágenes, 1 con el escudo y 16 con los distintos pedazos.

Ventajas:

- Solucionamos el problema de rendimiento computacional y tenemos una solución muy rápida.
- No incumpliríamos ningún principio fundamental de la ingeniería el software.

Desventajas:

- El problema de almacenamiento se empeora ya que hora en todo momento tendríamos 17 imágenes por cada escudo en memoria.

Conclusión:

Tras meditar teniendo en cuenta las tres alternativas, en primer lugar, se descartó la tercera ya que con esta el almacenamiento de la aplicación sufriría al tener tantas

imágenes por cada escudo. Y posteriormente se descartó también la segunda alternativa ya que, con esta en el caso de tener muchísimos jugadores de forma concurrente, la memoria también tendría problemas y podría llegar a fallar.

Finalmente se optó por la primera opción ya que, a pesar de no respetar un principio fundamental de la ingeniería del software, esta alternativa era la única que permitiría que la aplicación escalase si muchos usuarios la utilizasen.

Implementación

Una vez comenzado el desarrollo de la aplicación, van surgiendo ideas y problemas que no se habían estudiado en el proceso de diseño y que han sido tratados una vez comenzada la implementación, esta sección estará dedicada a comentar estas cuestiones que por diversas razones no fueron contempladas en el proceso de planificación y diseño.

Algoritmo de selección de pedazos

Una vez desarrollada la funcionalidad base del modo de juego, al hacer pruebas informales jugando varias partidas descubrí que hay muchos escudos que, al ser divididos en dieciséis pedazos, en muchos casos, los pedazos correspondientes a las cuatro esquinas no aportaban información como pista ya que solo mostraban un pedazo completamente blanco como se muestra en la siguiente imagen:

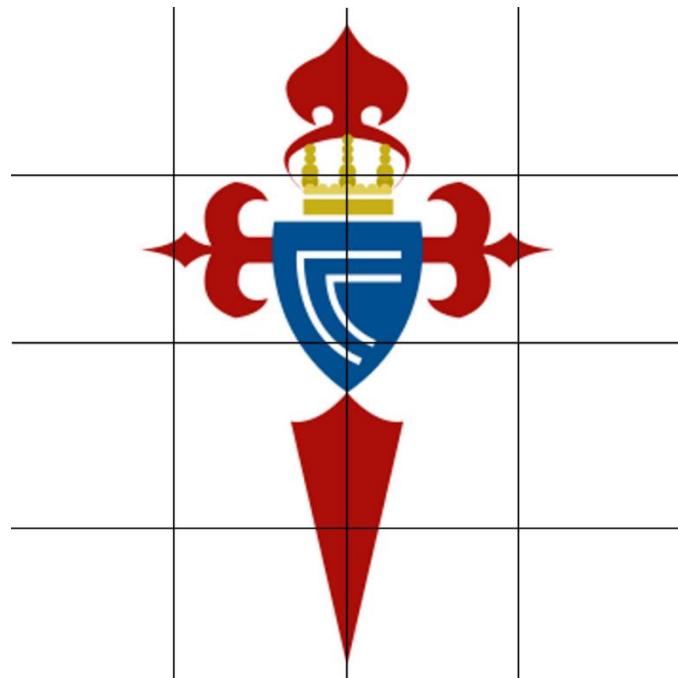


Figura 4.19: Ejemplo de un escudo con pedazos sin información

Como se puede ver en la imagen hay al menos 4 pedazo que no nos aportarían ninguna información si son seleccionados como pista. Para solucionar este problema surgieron dos posibles soluciones:

Alternativa 1: No permitir que la aplicación seleccione aleatoriamente un pedazo correspondiente a las esquinas en ningún caso.

Ventajas:

- Nos aseguraríamos de que todas las pistas diesen información sobre el escudo.

Desventajas:

- En los casos en los que un escudo si da información en las esquinas, nunca podríamos mostrar esa información como una pista.
- Además, en escudos como el de la imagen anterior, algunos de los pedazos laterales no muestran información y por lo tanto esta solución no sería del todo efectiva ya que utilizaría dichos pedazos como pistas.

Alternativa 2: Desarrollar un algoritmo que cada vez que se seleccione un pedazo compruebe el porcentaje de pixeles de cada color que hay en ese pedazo y que obligue a cambiar de pedazo, si un color representa más del 80% de los pixeles.

Ventajas:

- Cualquier pedazo que tenga un solo color o una cantidad casi inapreciable de pixeles de otro color, nunca será utilizado como pista.

Desventajas:

- Revisar la cantidad de pixeles de cada color que hay en un pedazo tiene un coste computacional y puede hacer que la aplicación tarde más tiempo en escoger un pedazo entre turno y turno.
- Puede que algunos pedazos tengan pedazos que a simple vista sean de un solo color, pero a nivel de pixeles haya más de una tonalidad y en estos casos el algoritmo puede fallar.

Alternativa 3: Introducir manualmente en los datos de la aplicación para cada escudo que pedazos no son válidos.

Ventajas:

- Tenemos en todo momento la decisión de que pedazos son válidos y cuáles no.

Desventajas:

- Hacer este proceso manualmente es muy costoso y repetitivo.

Conclusión:

Tras hacer varias pruebas con el algoritmo de la segunda alternativa, se observó que realmente el rendimiento de la aplicación no se veía reducido por el algoritmo de selección de pedazos y por lo tanto esta era la mejor opción ya que evaluaría en tiempo real que pedazos se pueden seleccionar y que pedazos no se pueden seleccionar.

Algoritmo de transformación de imágenes

Como comentamos en el apartado de decisiones de diseño, decidimos guardar las imágenes en el directorio de archivos del proyecto en contraposición de utilizar imágenes de internet y acceder a ellas por su URL.

Uno de los principales motivos para tomar esta decisión fue el hecho de que al tener almacenadas las imágenes en el directorio de archivos, podríamos trabajar con ellas para que todas tuviesen el mismo formato de archivo y las mismas dimensiones, así que implementé un script Python que transformase todas las imágenes de formato .png a .jpg para que todas dejaran de tener un fondo transparente y que el fondo siempre fuera blanco para ayudar al algoritmo de selección de pedazos. Además, con el script se redimensionaron todas las imágenes para que tuvieran unas dimensiones de 2000x2000 pixeles para que así todas se vieran igualmente proporcionadas. Dicho script se puede revisar en la carpeta “src/main/resources/utils” del proyecto.

De forma resumida lo que hace el script es recorrer todos los archivos de una carpeta y si el archivo tiene la extensión .png, hace una copia del archivo y la redimensiona, posteriormente se crea una imagen con un fondo blanco y sobre ella se pega la imagen que se había redimensionado. Finalmente se guarda la imagen en formato jpg. En el caso de que ocurra algún problema en el proceso de transformación, se notificará, pero el proceso continuará con el siguiente archivo.

Algoritmo de procesado de imágenes

Como anteriormente hemos comentado, decidí implementar un algoritmo para transformar las imágenes antes de utilizarlas en mi proyecto para que tuvieran el mismo formato y tamaño, pero al crear el módulo de administración y permitir que los administradores puedan incluir nuevas imágenes en la aplicación, tuve que desarrollar un nuevo algoritmo ya que el primero que desarrollé no era compatible al estar hecho en un lenguaje de programación diferente.

De forma resumida lo que hace este algoritmo es que cuando un administrador introduce una nueva imagen en el sistema, primero se comprueba el formato de la imagen y si es distinto de jpg o png, se rechaza la imagen, sin embargo, si se trata de uno de estos formatos, se redimensiona la imagen y se convierte a jpg. En este algoritmo a diferencia del script Python, en las imágenes png los pixeles transparentes pasarán a ser de color negro.

Integración del manual

El manual para usuarios debe quedar accesible en la aplicación para que los usuarios puedan consultarla en cualquier momento y la forma de implementarlo a dado lugar a dos posibles alternativas.

Alternativa 1: Implementar el manual directamente en el jsp.

Ventajas:

- El manual se integrará mejor en la interfaz de la aplicación.

Desventajas:

- Habría que utilizar la sintaxis de HTML para añadir el manual a la aplicación.

Alternativa 2: Utilizar un iframe para insertar el manual en forma pdf.

Ventajas:

- Se reutilizaría fácilmente el manual desarrollado para la memoria.

Desventajas:

- Es una solución menos estética y puede llegar a no integrarse bien en la interfaz.

Alternativa 3: Implementar el manual en formato MarkDown.

Ventajas:

- Esta solución se integrará mejor en la interfaz de la aplicación.

Desventajas:

- Más cantidad de trabajo que utilizar un iframe.
- Habría que utilizar la sintaxis MarkDown para añadir el manual a la aplicación.

Conclusión:

Tras pensarlo detenidamente, se optó por la primera alternativa ya que a pesar de tener un poco más de trabajo que reutilizar el manual ya desarrollado para la memoria, era la solución que mejor balanceada estaba entre cantidad de trabajo y resultado final.

Capítulo 5 Manual de usuario

En primer lugar, hay que tener en cuenta que en la aplicación existen dos tipos de usuarios, los jugadores y los administradores, dependiendo del tipo de usuario, se obtendrán más o menos permisos, por lo tanto, el manual se dividirá en dos partes, un manual con las funciones que pueden utilizar los jugadores y administradores y las funciones exclusivas para los administradores.

Manual para todos los usuarios

Una vez se accede a la aplicación web, la página principal que se puede observar es la siguiente:

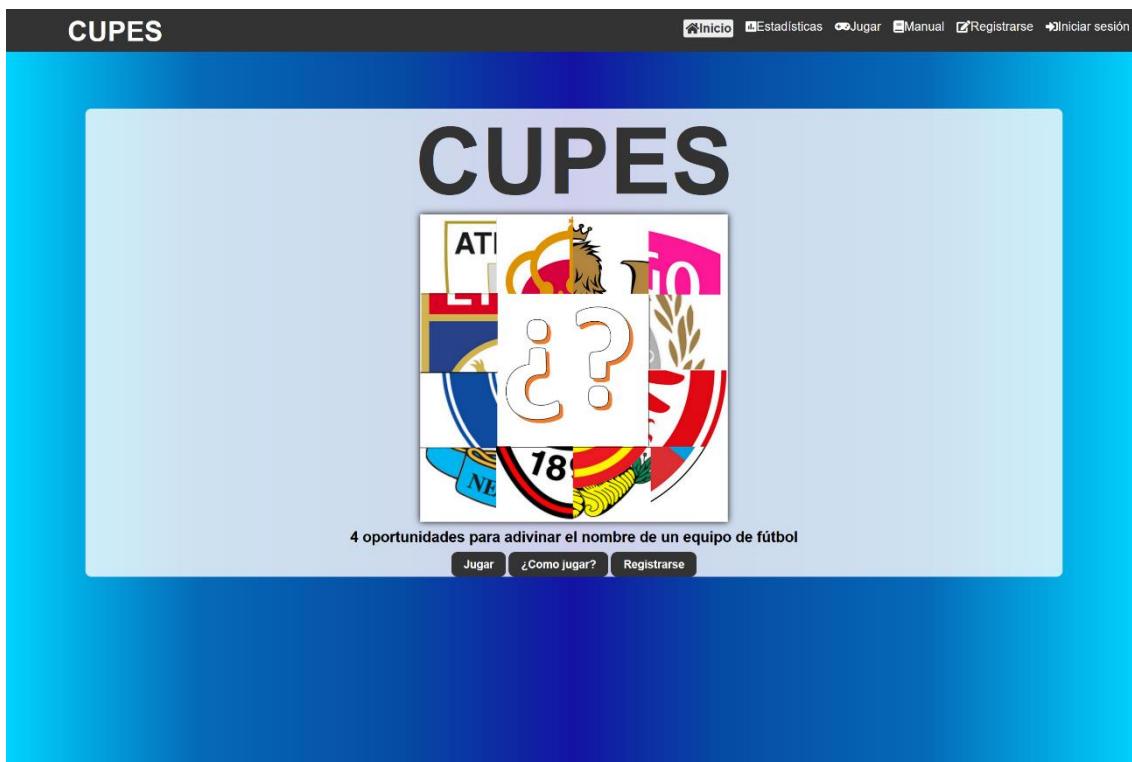


Figura 5.1: Página de bienvenida

Como se puede ver en tenemos una barra de navegación que siempre estará presente y podemos usarla como acceso rápido a las siguientes secciones:

- Inicio, nos lleva a la página principal, aunque para acceder a la pantalla principal también podemos pulsar sobre el logo de CUPES de la barra de navegación.
- Estadísticas, nos llevará al apartado de estadísticas, que dependiendo de si hemos iniciado sesión o no, nos mostrará nuestras estadísticas individuales o solo las estadísticas generales de la aplicación.

- Jugar, nos lleva al apartado donde podremos seleccionar el tipo de partida que queremos jugar.
- Manual, es la sección donde podremos consultar el manual en línea siempre que queramos.
- Si no hemos iniciado sesión, observamos una opción para acceder al formulario de registro y otra opción para acceder al formulario para iniciar sesión.
- En el caso de que hayamos iniciado sesión, nos aparecerá una opción para cerrar sesión.

Los jugadores podrán registrarse en cualquier momento a través de este formulario:

Figura 5.2: Formulario de registro

Todos los campos del formulario de registro son obligatorios y hay que tener en cuenta las siguientes restricciones:

- El nombre de usuario debe ser único, debe tener entre cinco y treinta caracteres y no puede contener espacios en blanco.
- La contraseña debe tener entre cuatro y treinta caracteres.
- El nombre debe tener entre tres y treinta caracteres.
- El email debe ser un correo valido y de una longitud menor a treinta caracteres.

Una vez registrados, lo usuarios podrán iniciar sesión:

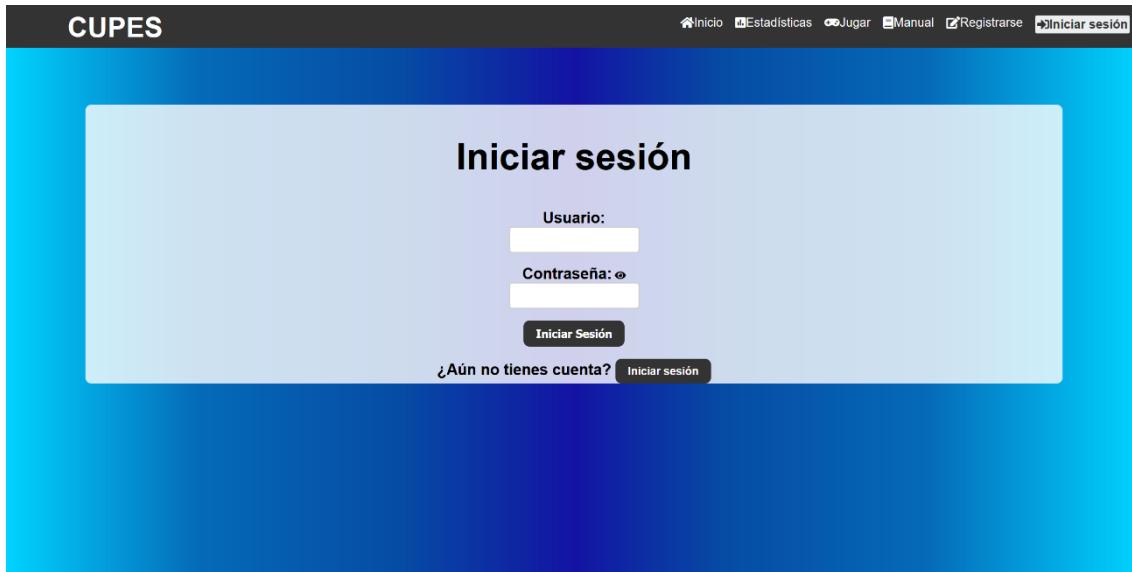


Figura 5.3: Inicio de sesión

En cualquier momento, los usuarios podrán consultar el apartado de estadísticas y en el caso de que hayan iniciado sesión, no solo se mostrarán las estadísticas generales, sino que se les mostrará sus propias estadísticas:

Estadísticas generales del modo un jugador	
Número de partidas:	8
Número de victorias:	7
Porcentaje de victorias:	87,5%
Victorias con un Intento:	6
Victorias con dos Intentos:	1
Victorias con tres Intentos:	0
Victorias con cuatro Intentos:	0
Media de intentos para ganar:	1,14

Estadísticas del usuario para el modo un jugador	
Número de partidas:	4
Número de victorias:	3
Porcentaje de victorias:	75%
Victorias en un Intento:	2
Victorias en dos Intentos:	1
Victorias en tres Intentos:	0
Victorias en cuatro Intentos:	0
Media de intentos para ganar:	1,33

Estadísticas de partidas en linea	
Número de partidas en linea:	1
Número de victorias del creador de la partida:	1
Número de victorias del jugador que se une a la partida:	0
Número de victorias perfectas(en el primer intento):	0
Media de intentos para ganar del creador de la partida:	4
Media de intentos para ganar del jugador que se une a la partida:	0

Estadísticas de partidas en linea del usuario	
Número de partidas:	1
Número de victorias:	1
Número de victorias perfectas:	0
Media de intentos para ganar :	4
Tasa de victorias del usuario:	100,00%
Tasa de victorias perfectas del usuario:	0,00%

Figura 5.4: Apartado de estadísticas

Tras iniciar sesión, los jugadores siempre podrán acceder a su perfil donde se les mostrarán sus datos y además sus propias estadísticas:

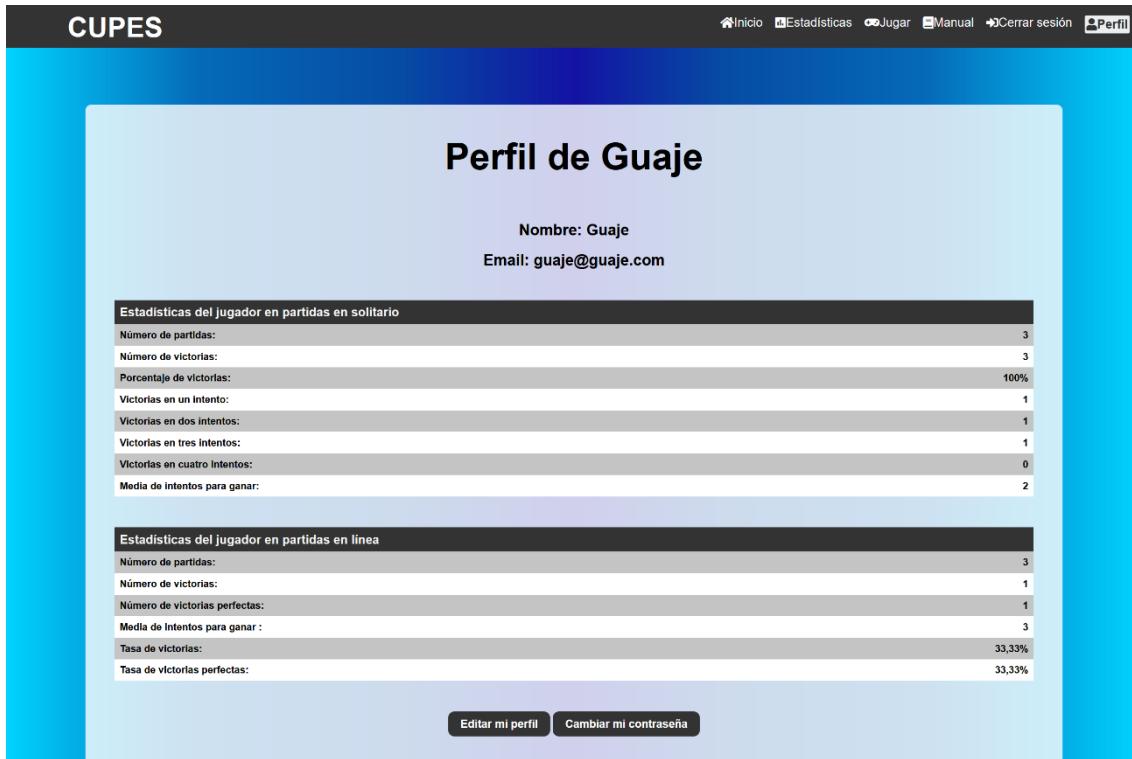


Figura 5.5: Perfil del usuario

Y desde el perfil se podrá acceder al formulario de edición del perfil:

The screenshot shows the "Editar mi perfil" (Edit profile) form. At the top, there is a dark header bar with the word "CUPES" on the left and navigation links on the right: Inicio, Estadísticas, Jugar, Manual, Cerrar sesión, and Perfil. Below the header is a light blue header section with the title "Editar mi perfil". Underneath this, the user's name "Nombre de usuario: Guaje" is displayed. The form contains fields for "Nombre:" (with value "Guaje") and "Email:" (with value "guaje@guaje.com"). At the bottom of the form is a "Editar" (Edit) button.

Figura 5.6: Edición del perfil

Al editar el perfil no podremos modificar nuestro nombre de usuario y el nombre y el email tendrán las mismas restricciones de longitud que en el formulario de registro.

Si queremos cambiar nuestra contraseña también podremos hacerlo desde el perfil y lo haríamos mediante el siguiente formulario donde deberemos introducir nuestra antigua contraseña y la cadena por la que queramos sustituirla teniendo en cuenta que debe tener entre cuatro y treinta caracteres.



Figura 5.7: Cambio de contraseña

Como la aplicación consta de varios modos de juegos, cada modo de juego tendrá su propio manual. En primer lugar, en la siguiente pantalla podemos seleccionar el modo de juego.

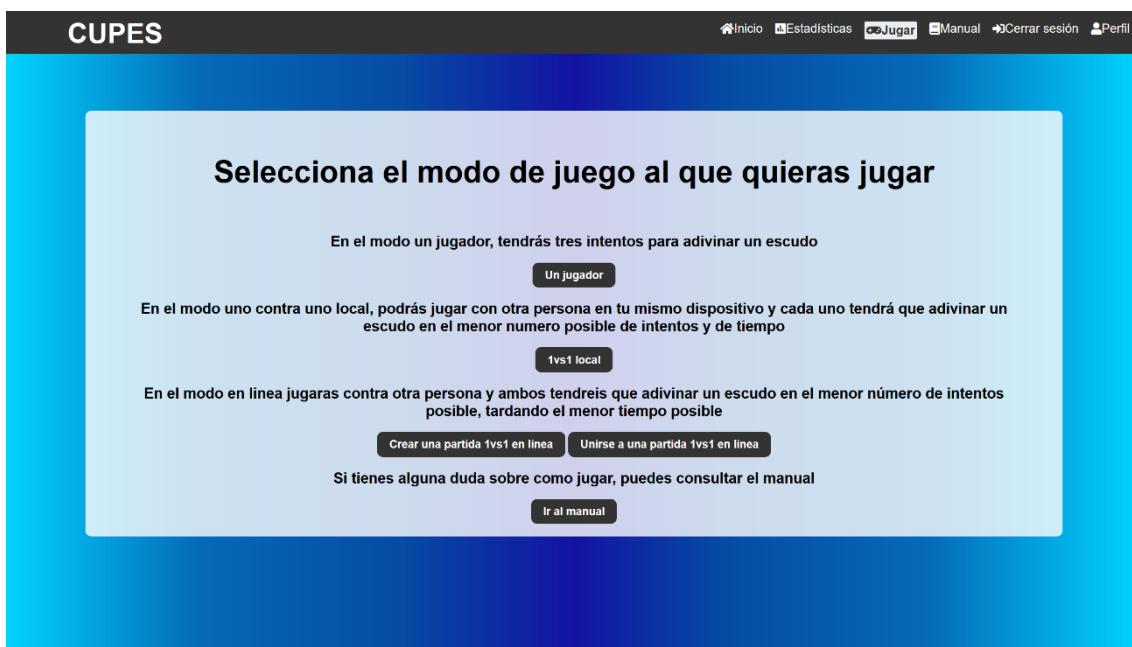


Figura 5.8: Seleccionar modo de juego

Manual del modo de juego un jugador:

Para jugar al modo de juego un jugador, no será necesario haber iniciado sesión ni haberse registrado, puesto que es un modo de juego rápido y cuya finalidad es practicar, aunque, si hemos iniciado sesión se guardarán nuestras estadísticas en este modo de juego. Una vez que en el selector de modos juegos pulsamos sobre jugar modo un jugador, la partida comenzará automáticamente y se nos mostrará la siguiente pantalla:

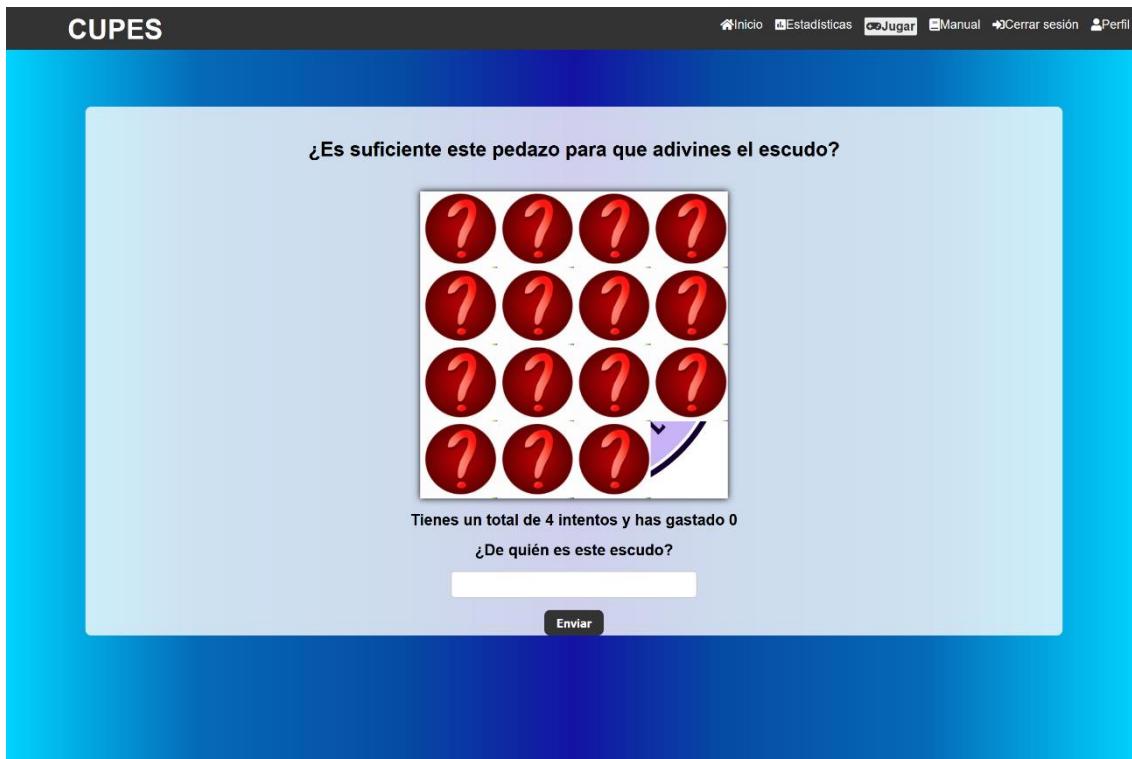


Figura 5.9: Partida en solitario comenzada

Como se puede observar, se muestra un escudo de un equipo completamente cubierto a excepción de un solo pedazo, que sería la pista de este intento , además, hay un mensaje que nos dice cuántos intentos podemos hacer y cuantos intentos llevamos. Bajo la imagen, se puede observar una entrada de texto, donde tendremos que introducir el nombre del club al que creemos que pertenece ese fragmento. Para introducir el nombre del club, el usuario comenzará a escribir y le saldrá un listado de opciones con el nombre completo de los clubs que en su nombre contienen la cadena de texto escrita por el usuario. Una vez en la lista de opciones se encuentre el club del que el usuario cree que es el escudo, el usuario deberá escogerlo en la lista de opciones para poder enviar el nombre completo del club sin errores. Con el nombre seleccionado, el usuario puede enviar su intento.

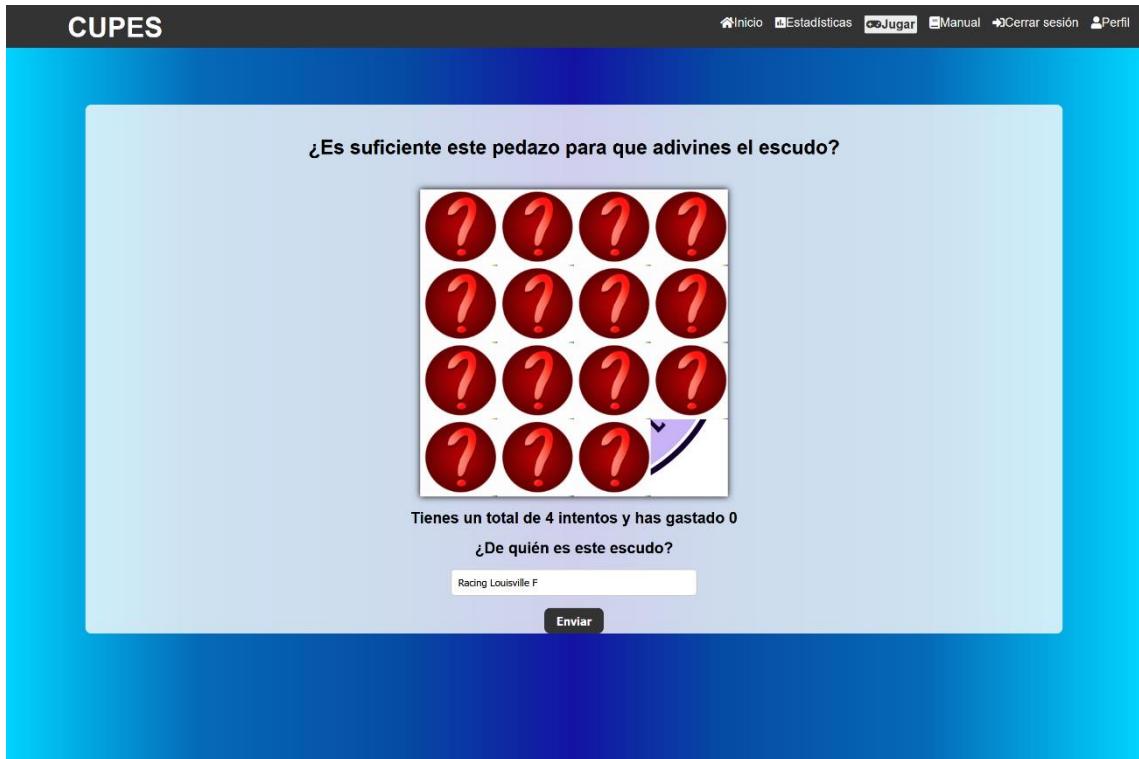


Figura 5.10: Ejemplo de envío de un intento

Según si ha acertado o no, pueden ocurrir dos cosas, si el usuario no ha acertado, habrá gastado un intento y se lo redireccionará a la misma página, donde se destapará un nuevo pedazo para que pueda volver a probar con la nueva pista:

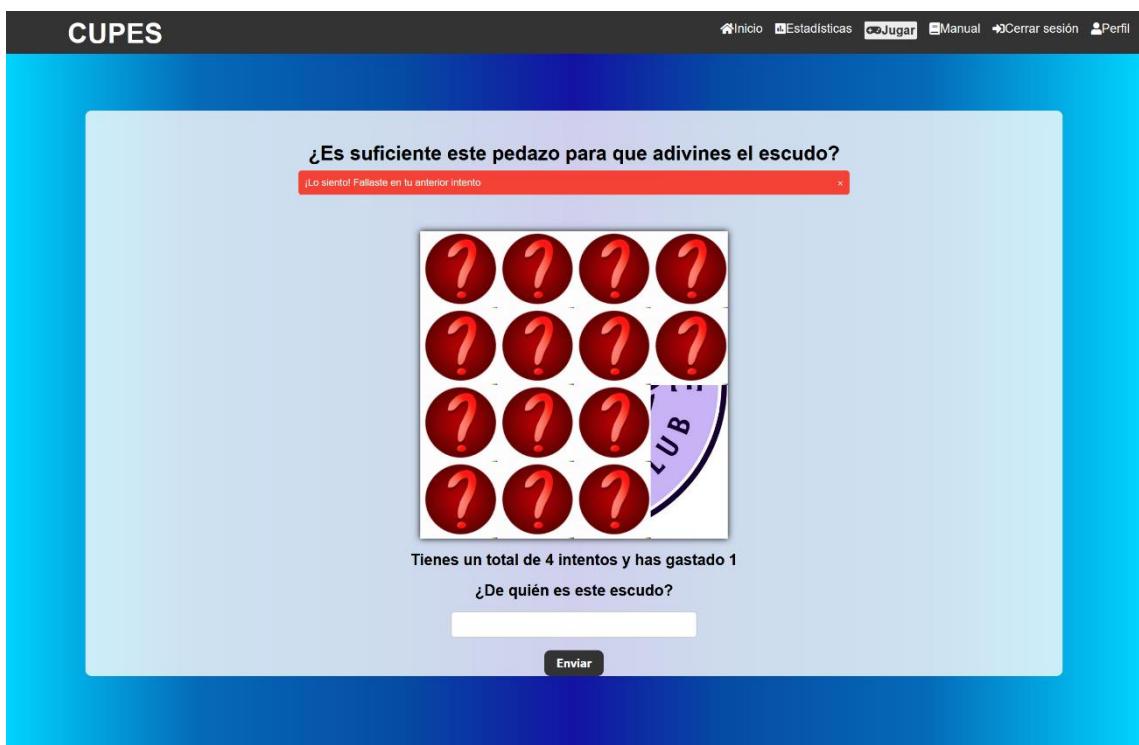


Figura 5.11: Ejemplo de fallo en solitario

Sin embargo, si el usuario ha acertado, se lo redirigirá a la siguiente pantalla, que le indicará que ha ganado y le mostrará el escudo completo del equipo.

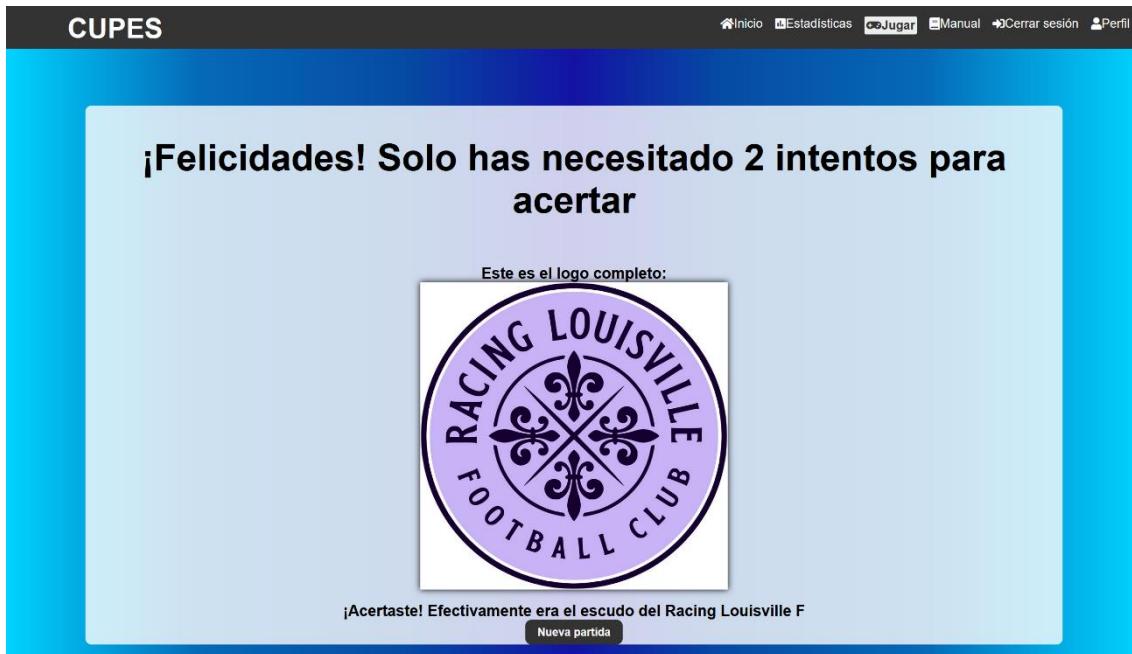


Figura 5.12: Victoria en solitario

Si un jugador gasta todos sus intentos y no consigue adivinar a qué club pertenece el escudo, también será redireccionado a una página con el resultado de la partida donde se le indicará que ha perdido y se le mostrará el escudo completo para que, de cara a una próxima partida lo conozca.

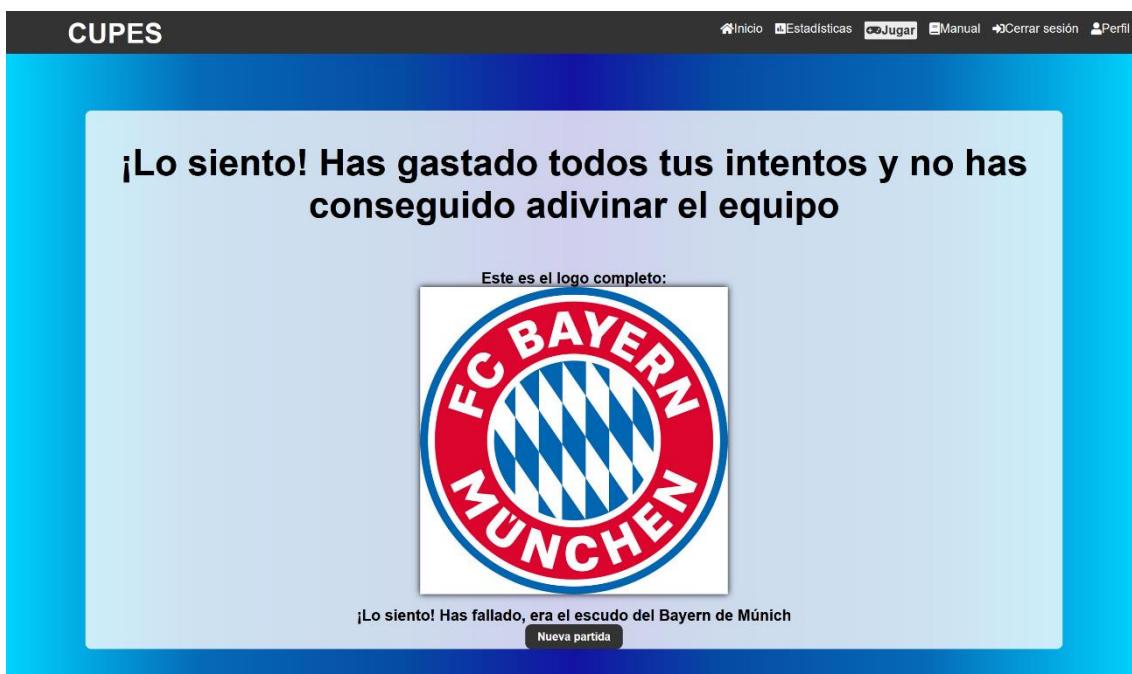


Figura 5.13: Derrota en solitario

Manual del modo de juego 1vs1 local:

Para jugar una partida 1vs1 local, primero tendremos que dirigirnos al selector de modos de juego que anteriormente hemos comentado y pulsar sobre el botón 1vs1 local, es importante recordar que para jugar a este modo tampoco hace falta haber iniciado sesión y que no hay apartado estadístico para este modo. Tras pulsar el botón observaremos la siguiente pantalla:

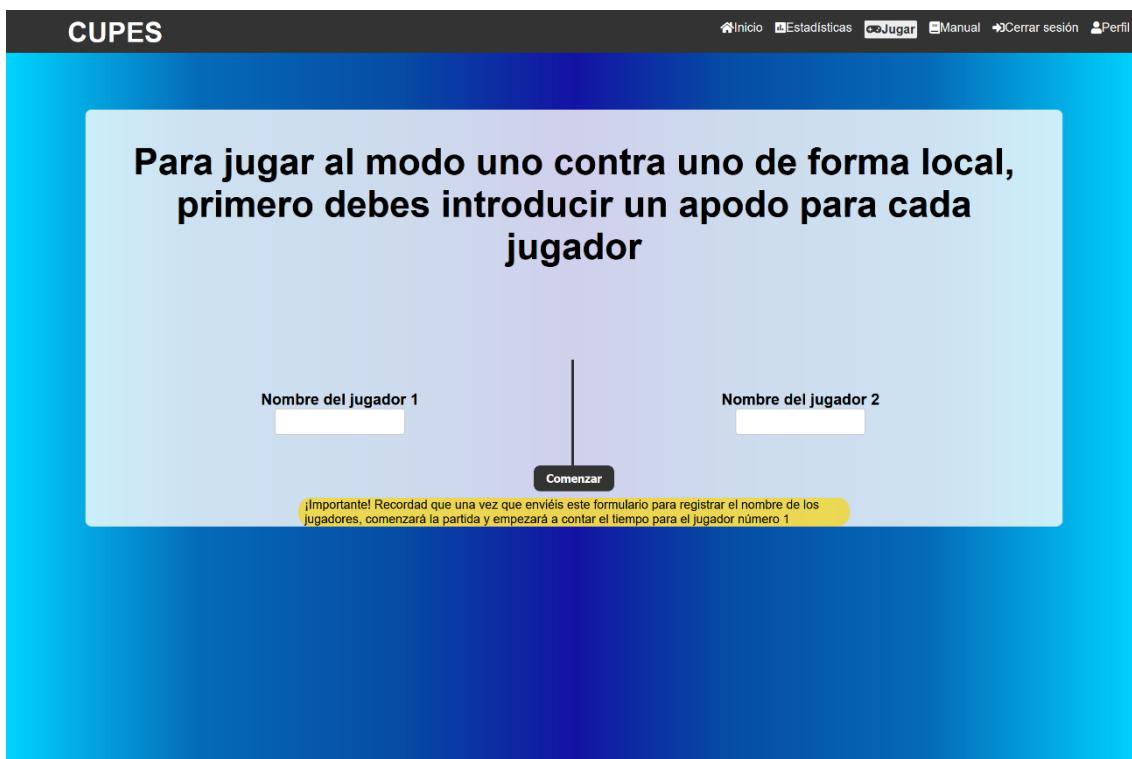


Figura 5.14: Crear partida uno contra uno local

Para llenar el formulario con el nombre de los dos jugadores que van a participar, hay que tener en cuenta las siguientes restricciones:

- Ninguno de los nombres puede estar vacío.
- Los nombres no pueden ser iguales.
- Ambos nombres deben tener una longitud de entre cuatro y treinta caracteres.

Una vez que se rellene el anterior formulario, será el turno del jugador 1 y, por lo tanto, este jugador deberá tratar de adivinar el escudo que le ha tocado. Como se puede ver en la siguiente pantalla, se indica el nombre del jugador que debe contestar actualmente y el número de intentos que ha gastado junto al número de intentos total que tiene.

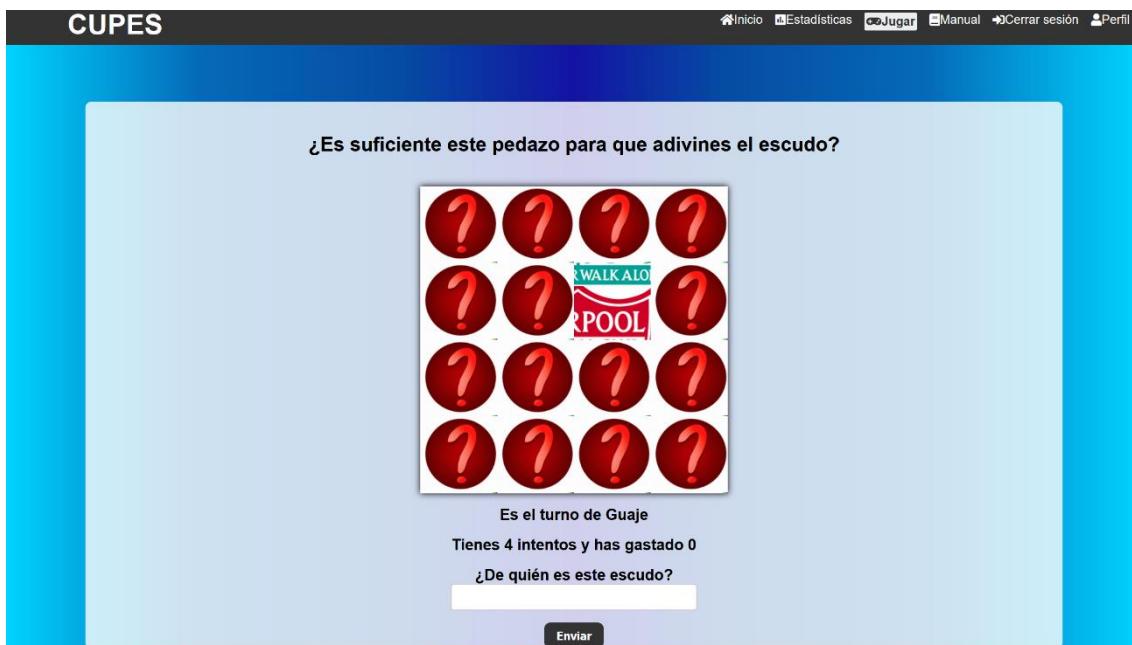


Figura 5.15: Partida uno contra uno local

A partir de este momento el jugador 1 utilizará sus intentos y podrán ocurrir dos cosas, que el jugador consiga acertar el escudo o que el jugador gaste sus cuatro intentos y no consiga adivinar el escudo, por lo tanto, no podrá ganar, como mucho podrá empatar en el caso de que el jugador 2 tampoco sea capaz de adivinar su escudo.

Cuando el jugador falle se mostrará un nuevo pedazo y se le notificará de que no ha acertado:

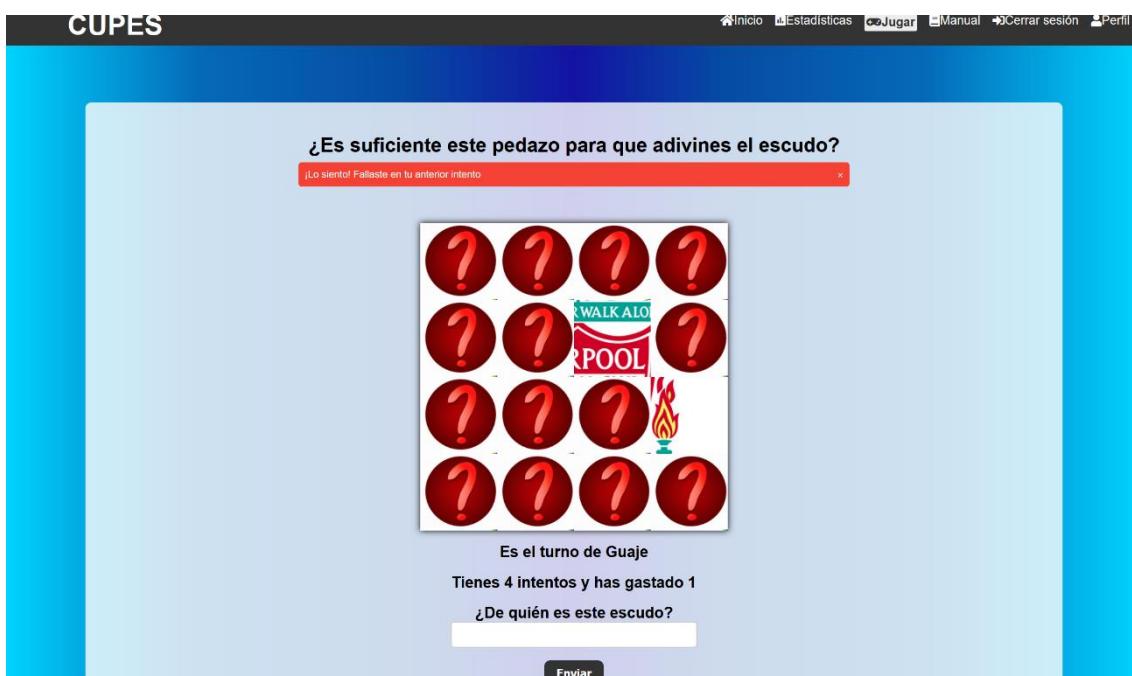


Figura 5.16: Fallo en partida uno contra uno local

Tanto si acierta como si gasta sus intentos, pasará a ser el turno del jugador 2 y se podrá observar que la frase es el turno de “x”, ha cambiado.

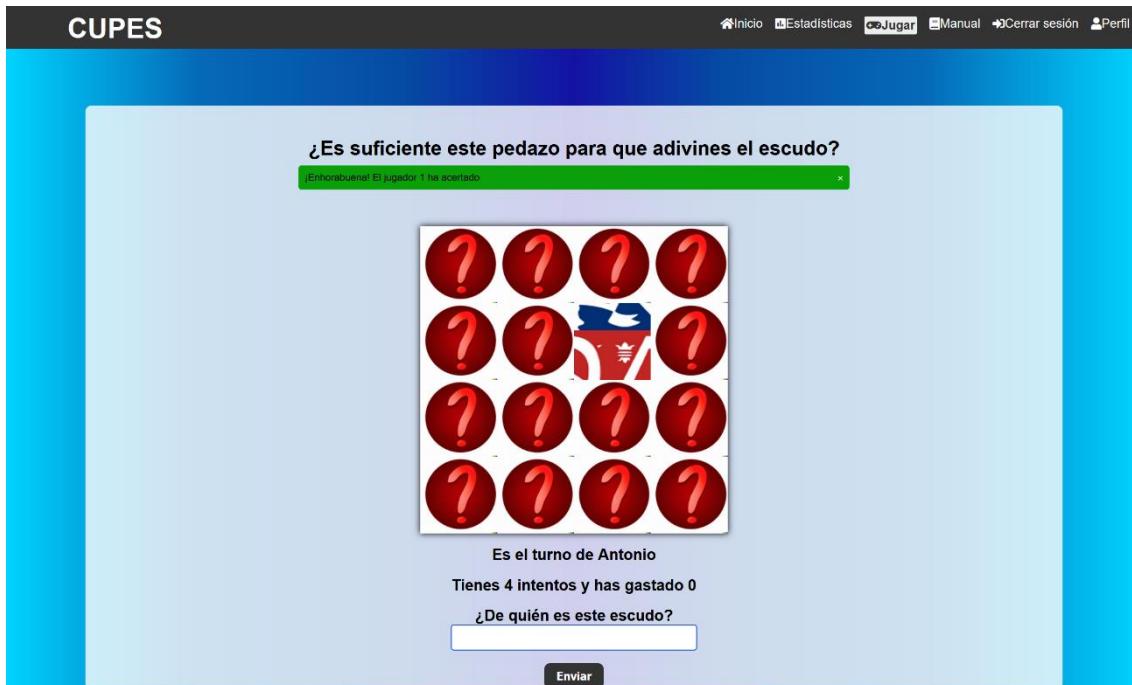


Figura 5.17: Acierto en partida uno contra uno local

El escudo de ambos jugadores se escoge de forma simultánea y existe la posibilidad de que el escudo de ambos jugadores sea el mismo, aunque la probabilidad sea muy pequeña.

Una vez que comience el turno del jugador 2, este podrá empezar a intentar adivinar su escudo y cuando adivine el escudo o gaste sus intentos, la aplicación determinará el ganador de la siguiente forma:

- Si solo un jugador adivina el escudo este ganará.
- Si ambos jugadores no adivinan su escudo empatarán.
- Si los dos jugadores adivinan el escudo, ganará quien menos intentos haya necesitado, en caso de empate en el número de intentos, el criterio para determinar el ganador será el tiempo, resultando como ganador quien menos tiempo haya necesitado. En caso de empate en tiempo e intentos, el resultado de la partida será un empate.

Con la partida terminada y el ganador de esta determinado, se mostrará la siguiente pantalla donde podremos ver las estadísticas de la partida.

El ganador ha sido Guaje

Guaje ha acertado y ha necesitado 2 intentos y ha tardado 14 segundos

Antonio ha acertado y ha necesitado 2 intentos y ha tardado 21 segundos

Este es el escudo de Guaje



Este es el escudo de Antonio



[Nueva partida](#)

Figura 5.18: Resultado partida uno contra uno local

Manual del modo de juego en línea:

Para jugar una partida en línea en primer lugar tendremos que habernos identificado en el sistema previamente y tras identificarnos, tendremos dos opciones, podremos crear una partida o unirnos a una que ya este creada y no haya empezado. Para cualquiera de las opciones, tendremos que ir a la página de modos de juego y ahí podremos seleccionar la opción que más nos convenza. Si decidimos crear una partida, la aplicación la creará y nos llevará al lobby.

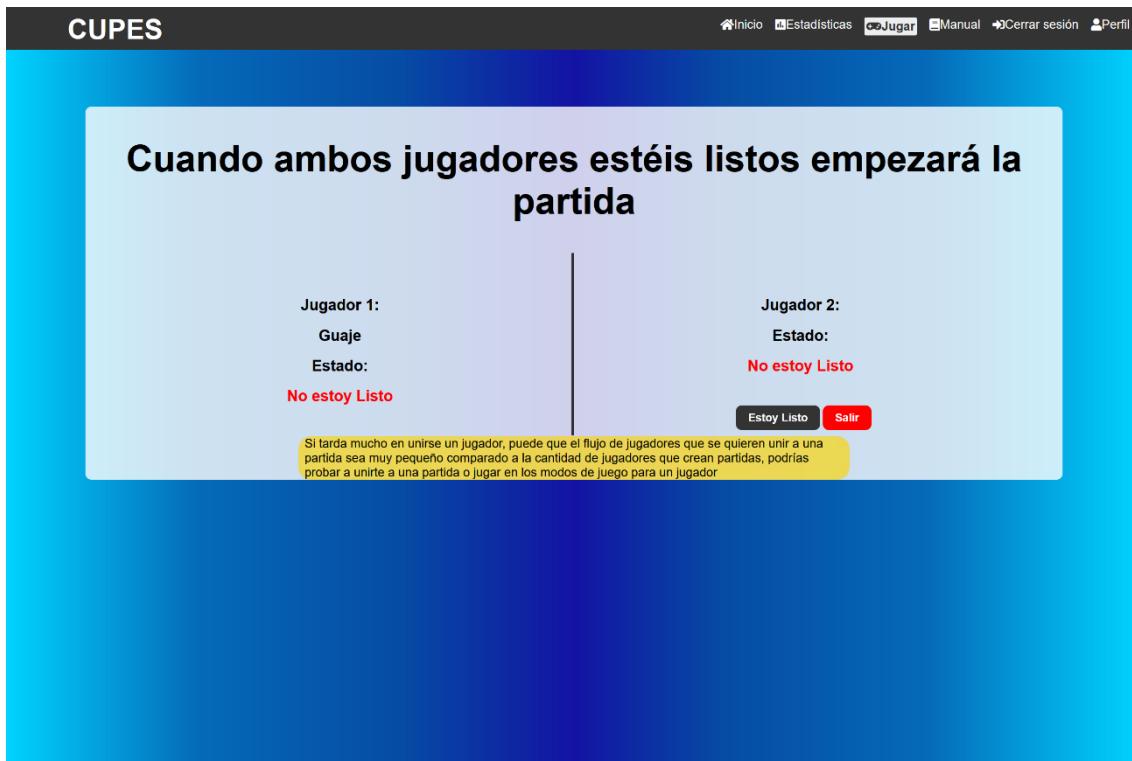


Figura 5.19: Creación partida en línea

Sin embargo, si decidimos unirnos a una partida, puede que tengamos que esperar y mientras esperamos aparecerá el siguiente mensaje en la pantalla, aunque si hay partidas en las que todavía no se haya unido un segundo jugador, la pantalla no aparecerá, sino que directamente seremos direccionados al lobby de una partida. En la pantalla de espera, tendremos un pasatiempo que constará de un juego de memoria en el que le tendremos que dar la vuelta a las siguientes cartas de dos en dos para encontrar las parejas de escudos.



Figura 5.20: Unirse a partida en línea

Es importante tener en cuenta que tanto si queremos crear una partida como si queremos unirnos a una que puede que la aplicación tarde en conseguir hacer el emparejamiento ya que si alguien crea una partida, es necesario que alguien más quiera unirse a una partida, por lo que si todos los usuarios de la aplicación deciden crear una partida, no habrá nadie que se une a esas partida y por lo tanto estas nunca comenzarán, por lo se recomienda encarecidamente a los usuarios que si crean una partida y nadie se une, prueben a unirse a una partida, porque puede que el flujo de jugadores que crean partidas sea mayor que el flujo de jugadores que se unen a partidas. En el caso opuesto también se recomienda que, si se busca unirse a una partida y tarda mucho, se pruebe a crear una partida.

Una vez que un jugador haya creado la partida y otro se haya unido, aparecerá el siguiente lobby, donde ambos jugadores podrán decidir cuando están listos y podrán abandonar la partida antes de comenzar. Si la partida la abandona el jugador que la crea, esta partida desaparecerá y nunca se jugará, sin embargo, si quien abandona es el jugador que se ha unido a ella, el emparejamiento seguirá funcionando y se podrá unir otro jugador.

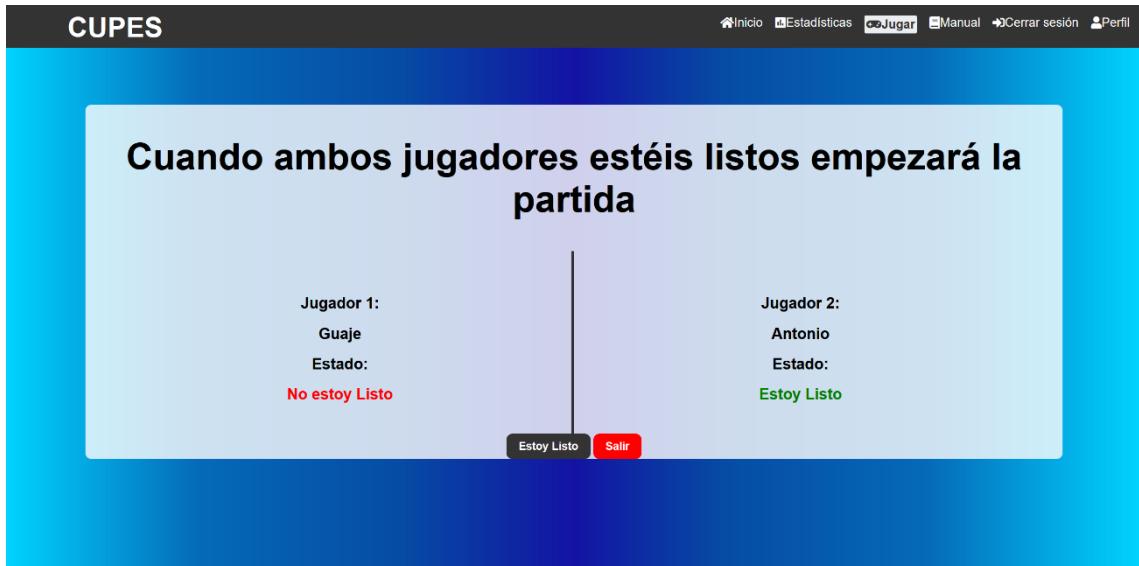


Figura 5.21: Lobby de partidas en línea

Una vez que ambos jugadores estén listos, la partida habrá comenzado y ambos serán redireccionados automáticamente a la página donde se jugará la partida, en la página se les indicará el número de escudos que llevan y el número de intentos, mientras que se muestra el escudo cubierto con las pistas de la partida.



Figura 5.22: Partida en línea

Cada vez que el jugador acierte se mostrará el siguiente escudo hasta que adivine los tres:



Figura 5.23: Acierto en partida en línea

Cada vez que falle, se mostrará un pedazo más hasta que se hayan utilizado 4 intentos, si tras cuatro intentos no se ha conseguido acertar el escudo, se pasará al siguiente.

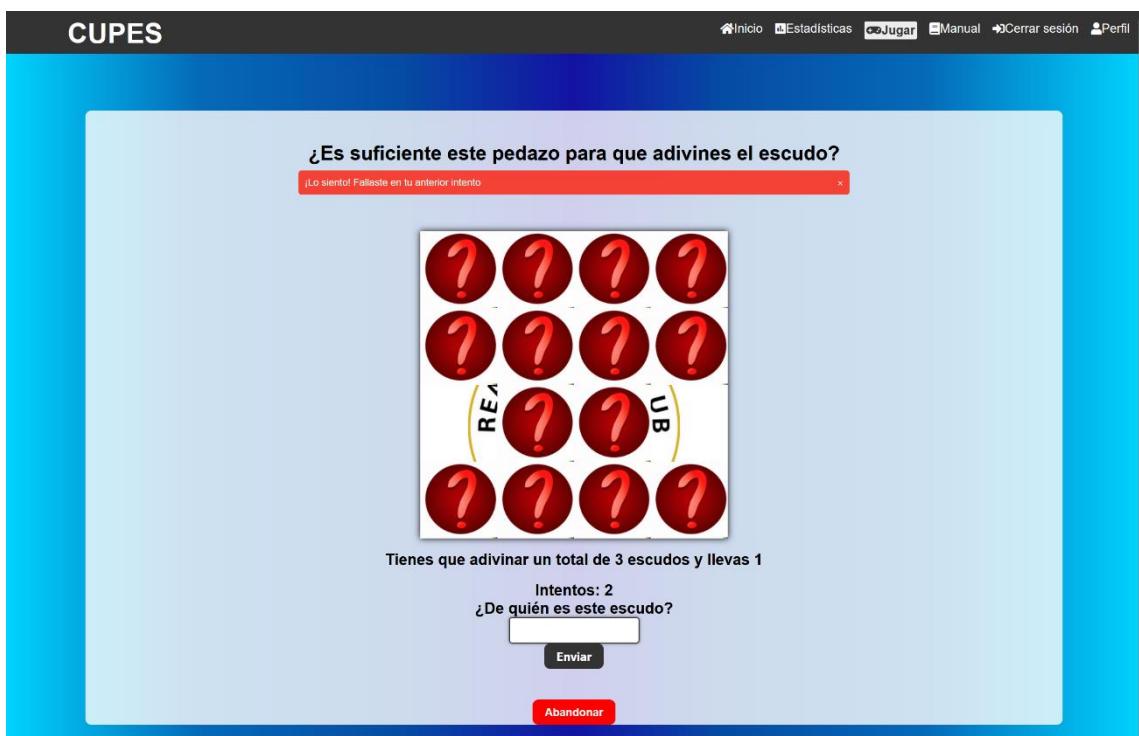


Figura 5.24: Fallo en partida en línea

Una vez que un jugador gaste todos sus intentos o adivine los tres escudos, pasará a una página donde tendrá que esperar hasta que el otro jugador termine, para que la espera sea más amena, en esta pantalla también se ha incluido el pasatiempo donde hay que levantar las cartas de dos en dos para conseguir parejas.

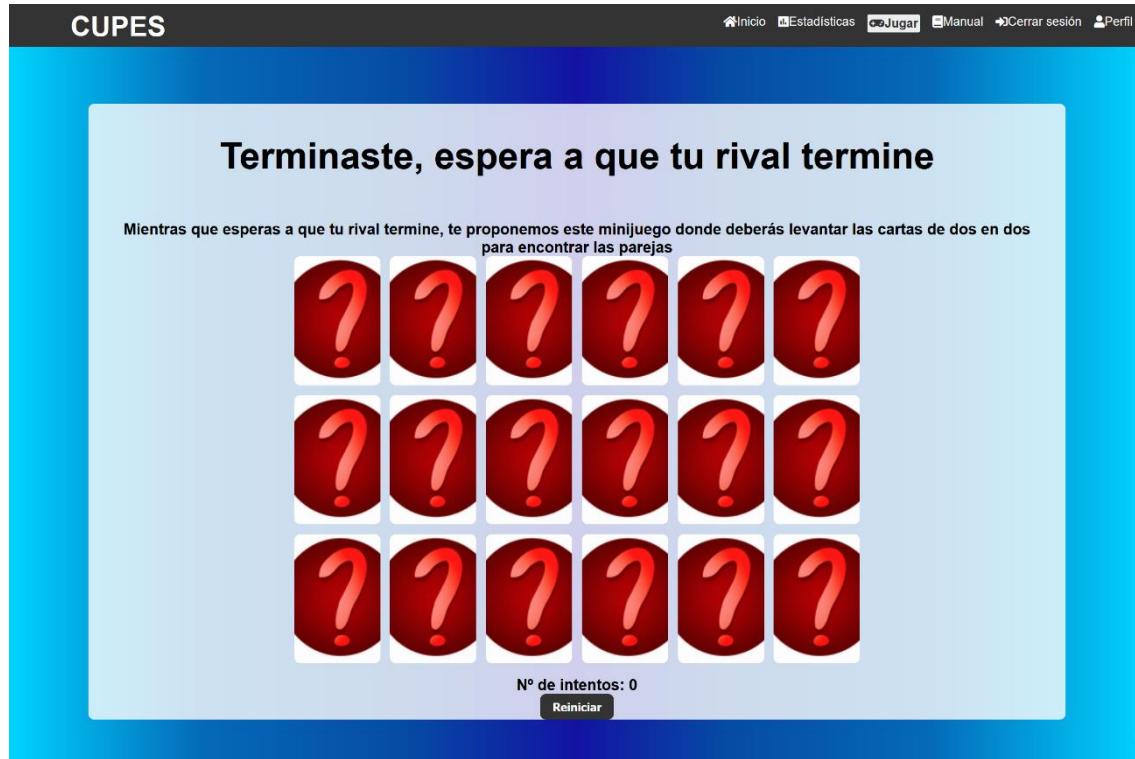


Figura 5.25: Minijuego esperando terminar partida en línea

Cuando ambos jugadores terminen, el sistema determinará el ganador de la partida de la siguiente forma:

- Gana quien más escudos acierte.
- En caso de empate a aciertos, gana quien menos intentos haya necesitado.
- En caso de empate a intentos, gana quien menos tiempo haya tardado.
- En caso de empate en tiempo y aciertos, el resultado de la partida será empate.



Figura 5.26: Resultado partida en línea

En cualquier momento de la partida, un jugador podrá decidir abandonarla pulsando el botón rojo de abandonar partida, lo que provocará automáticamente su derrota y la victoria del otro jugador que será redirigido automáticamente a la pantalla de resultado de la partida.



Figura 5.27: Resultado por abandono en línea

Manual de administración

Si iniciamos sesión con una cuenta que tenga permisos de administrador, podremos acceder a todas las funcionalidades explicadas en el manual general y adicionalmente se podrá acceder a las funciones del módulo de administración, para obtener permisos de administrador, podemos iniciar sesión con las siguientes credenciales: ‘admin’ ‘1111ñ@a?’.

Tras iniciar sesión, podremos observar que en la barra de navegación a aparecido una nueva opción.

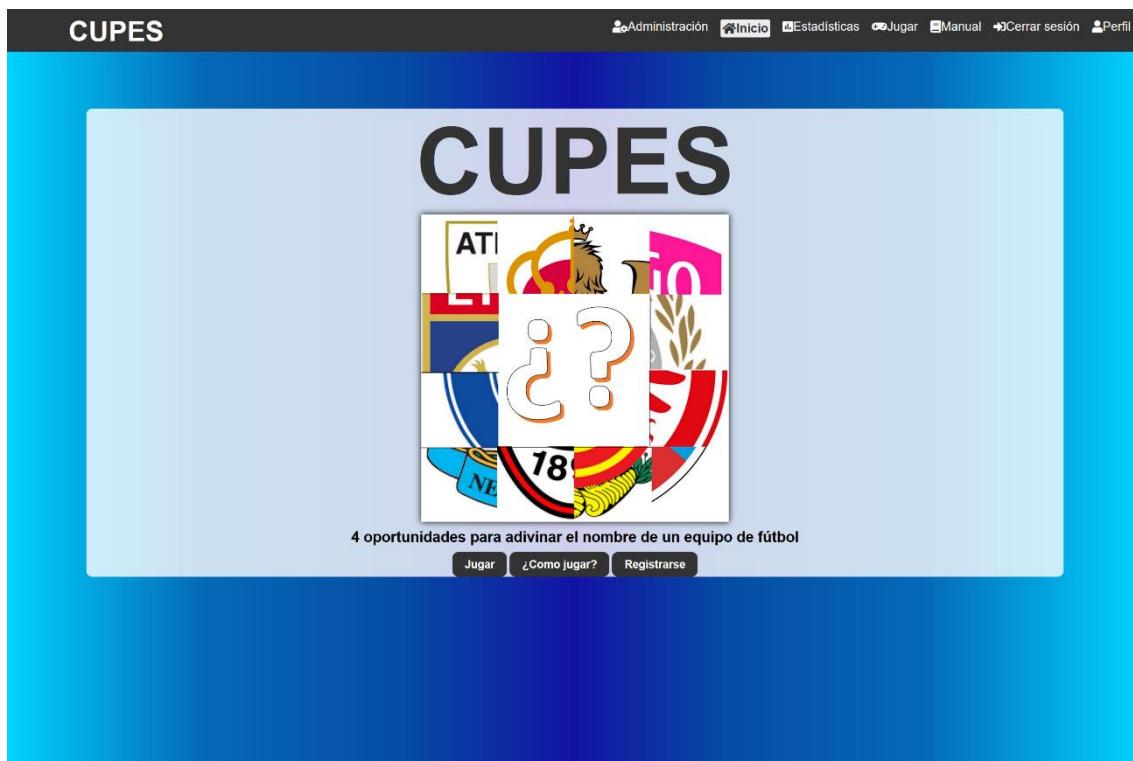


Figura 5.28: Pagina de bienvenida para administradores

Si pulsamos sobre administración, nos llevará a la siguiente sección:



Figura 5.29: Opciones módulo de administración

El apartado de administrar jugadores nos permitirá ver un listado paginado de todos los jugadores registrados en la aplicación, permitiéndonos deshabilitar sus cuentas para que no puedan acceder a la aplicación si lo creemos conveniente. Igual que podemos deshabilitarlos, en cualquier momento podemos volver a habilitarlos.

Nombre de usuario	Mail	Estado	Deshabilitar	Habilitar
Guaje	guaje@guaje.com	●	■	▶
Antonio	antonio@antonio.com	●	■	▶
Mera	mera@mera.com	●	■	▶
admin	admin@admin.com	●	■	▶

Figura 5.30: Listado de usuario para el administrador

Como se puede observar, también contamos con un buscador que facilitará la búsqueda de jugadores cuando la aplicación crezca y haya muchos usuarios.

Si pulsamos sobre sus nombres, podemos acceder a sus perfiles para consultar todos sus datos y sus estadísticas:

Estadísticas del jugador en partidas en solitario	
Número de partidas:	6
Número de victorias:	4
Porcentaje de victorias:	66,67%
Victorias en un intento:	2
Victorias en dos Intentos:	2
Victorias en tres intentos:	0
Victorias en cuatro intentos:	0
Media de intentos para ganar:	1,5

Estadísticas del jugador en partidas en línea	
Número de partidas:	3
Número de victorias:	3
Número de victorias perfectas:	0
Media de intentos para ganar :	3
Tasa de victorias:	100,00%
Tasa de victorias perfectas:	0,00%

Figura 5.31: Perfil de un usuario desde la vista del administrador

Como hemos iniciado sesión como administradores si accedemos al perfil de alguien, tendremos un botón para exportar sus estadísticas de las partidas en solitario en formato .json. Esto nos permitirá estudiar las estadísticas de los jugadores.

Además, podremos acceder a un listado similar al de los jugadores, pero de las imágenes registradas en la aplicación, este listado también está paginado y tendrá un buscador. En este caso si deshabilitamos una imagen esta ya no podrá ser seleccionada en las siguientes partidas de todos los modos de juego.

Estas son las imágenes que pueden aparecer en la aplicación			
Nombre	Imagen	Deshabilitar/Habilitar	Ver datos
Gothan F		x	v
Orlando Pride		x	v
Washington Spirit		x	v
Racing Louisville F		x	v
Houston Dash		x	v
Kansas City Current		x	v
Chicago Red Stars		x	v

Páginas: [1](#) [2](#) [3](#) [4](#) [5](#) [6](#) [7](#) [8](#) [9](#) [10](#) [11](#) [12](#) [13](#)

Figura 5.32: Listado de imágenes para el administrador

Desde este listado si en la columna Deshabilitar/Habilitar nos sale una “x” es porque esta imagen está habilitada y la podemos deshabilitar pulsando sobre la “x”, en el caso contrario, es decir, si la imagen está deshabilitada nos saldrá un “v” y pulsando sobre dicha “v” habilitaremos la imagen de nuevo.

Si pulsamos sobre el ojo, accederemos a los datos de una imagen:



Figura 5.33: Detalles de una imagen

En primer lugar, podremos observar la imagen y los 16 trozos que la componen, debajo podemos ver los datos de la imagen y alguna estadística, también tenemos un botón para deshabilitar/habilitar la imagen, una opción para exportar todos los datos de las partidas en solitario en las que ha aparecido la imagen a un archivo .json y por último una muestra de cuales de los trozos de la imagen son válidos para ser pistas en una partida (en color verde) y los trozos que no son válidos ya que no aportan suficiente información como pistas (en rojo).

La aplicación también proporciona un formulario para que los administradores puedan añadir nuevas imágenes a la aplicación y que estas puedan salir en las partidas.

Añade aqui la imagen No file selected.

¿Cual es el nombre completo de este equipo?

¿Cual es el país de este equipo?

Selecciona la categoría del equipo:

Selecciona el género del equipo:

¿El escudo muestra el nombre del equipo?

¿El escudo muestra las iniciales del equipo?

¿El escudo muestra el año de fundación del equipo?

¿Deseas habilitar el escudo para que aparezca en partidas?

Figura 5.34: Formulario para añadir una nueva imagen

Además, la aplicación tiene un apartado de estadísticas de las imágenes para que los administradores puedan sacar conclusiones de que tipo de escudos reconocen mejor los usuarios de la aplicación. Desde este apartado también se pueden exportar datos de las partidas en solitario a un archivo .json permitiendo que los administradores escojan en qué periodo de fechas quieren extraer las estadísticas.

Estadísticas para el administrador:

Estadísticas por imágenes del modo un jugador	Estadísticas generales	Con un intento	Con dos intentos	Con tres intentos	Con cuatro intentos
Porcentaje de acierto en escudos de España:	100	80	20	0	0
Porcentaje de acierto en escudos de Alemania:	100	100	0	0	0
Porcentaje de acierto en escudos de Francia:	0	0	0	0	0
Porcentaje de acierto en escudos de Italia:	100	100	0	0	0
Porcentaje de acierto en escudos de Inglaterra:	0	0	0	0	0
Porcentaje de acierto en escudos de Estados Unidos:	33,33	0	33,33	0	0
Porcentaje de acierto en escudos masculinos:	87,5	75	12,5	0	0
Porcentaje de acierto en escudos femeninos:	33,33	0	33,33	0	0
Porcentaje de acierto en clubes de primeras divisiones:	62,5	37,5	25	0	0
Porcentaje de acierto en clubes de segunda division:	100	100	0	0	0
Porcentaje de acierto en escudos con Nombres:	75	50	25	0	0
Porcentaje de acierto en escudos sin Nombres:	66,67	66,67	0	0	0
Porcentaje de acierto en escudos con Iniciales:	75	50	25	0	0
Porcentaje de acierto en escudos sin Iniciales:	66,67	66,67	0	0	0
Porcentaje de acierto en escudos con años:	75	50	25	0	0
Porcentaje de acierto en escudos sin años:	66,67	66,67	0	0	0

Selecciona un rango de fechas para exportar las estadísticas de las partidas

Fecha de inicio:

Fecha de fin:

Figura 5.35: Estadísticas generales de las imágenes

Capítulo 6 Pruebas

En el desarrollo de software es tan importante construir el software como asegurarnos de que funcione correctamente. Cuando hablamos de funcionar correctamente nos referimos a que hay que asegurarse de que el código siempre se comporta como esperamos que se comporte. Para esto, se desarrollan las pruebas.

La realización de pruebas es una práctica fundamental en los proyectos software para encontrar errores en el código desarrollado, las pruebas no solo deben ser desarrolladas, sino que hay que ejecutarlas periódicamente e incluso automatizar su ejecución para que, si alguna funcionalidad deja de funcionar correctamente, nos podamos dar cuenta gracias a los reportes de fallo de las pruebas. A lo largo de este capítulo comentaremos como se han hecho las pruebas de este proyecto y como se ha automatizado su ejecución.

En concreto en este proyecto se han desarrollado un total de 195 pruebas de distintos tipos como podrían ser los siguientes:

- **Tipo de prueba:** Pruebas unitarias.

Descripción: Son aquellas que evalúan individualmente pequeñas unidades de código como podrían ser funciones o métodos, garantizando así que las unidades de código que son objeto de estas pruebas realizan las tareas correctamente y producen los resultados esperados. El objetivo principal de estas pruebas es asegurar la calidad de cada componente por separado antes de ser integrado en el sistema.

Beneficios: Este tipo de pruebas tienen ventajas como la temprana detección de errores, además, no solo nos ayudan a detectar errores, sino que también nos ayudan a localizar que parte del código es la que produce los errores.

Uso en el proyecto: En el proyecto se han utilizado este tipo de pruebas para asegurar la calidad de algunos métodos importantes para la aplicación, comprobando su correcto funcionamiento. Un ejemplo pueden ser las pruebas sobre la función que comprueba si una porción de una imagen tiene suficiente cantidad de información como para ser considerada una porción válida como pista. En este caso, se ha tomado la siguiente imagen:

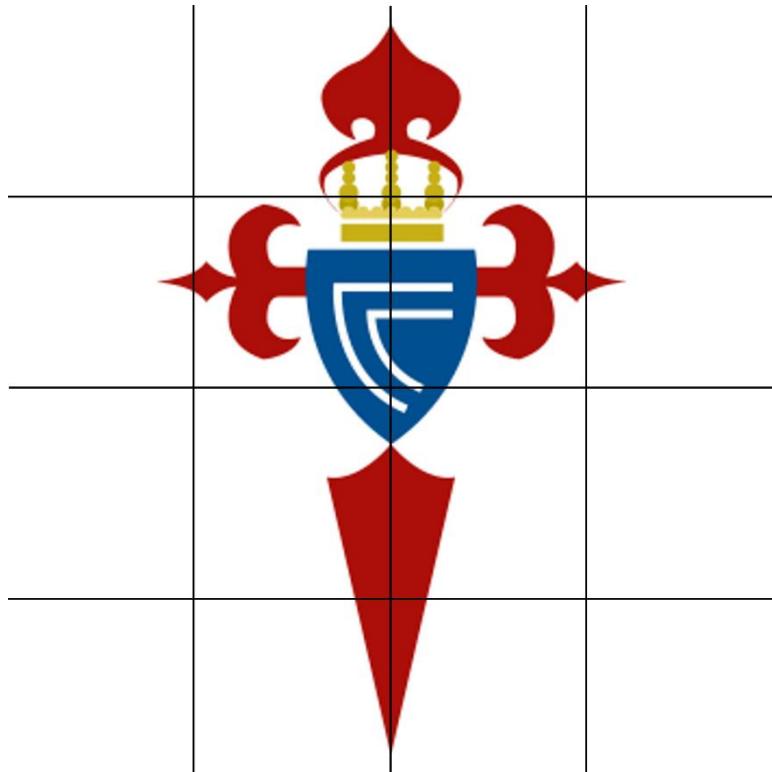


Figura 6.1: Escudo con pedazos no validos

Como se puede observar, hay ocho pedazos que no proporcionan suficiente información porque prácticamente solo tienen color blanco, pues se han hecho pruebas para asegurarnos de que esos ocho pedazos para la aplicación no son válidos como pista y sin embargo los otros ocho restantes sí que son válidos.

Otro ejemplo de pruebas unitarias en el proyecto podría ser cuando se comprueba que individualmente cada método que calcula una estadística y se comprueba que cada uno calcula correctamente su estadística para unos datos de prueba.

- **Tipo de prueba:** Pruebas funcionales.

Descripción: Son aquellas que evalúan si un sistema cumple con los requisitos y especificaciones funcionales para los que fue diseñado. Estas pruebas se centran en una cantidad de código mayor y cada prueba de este tipo se suele corresponder con una función y en ellas se suelen simular acciones que podría hacer el usuario para comprobar si el sistema se comporta como se espera.

Beneficios: Nos ayudan a validar los requisitos y nos ayudan a detectar que funcionales fallan, aunque no nos proporcionan tanta información sobre la localización del fallo como las pruebas unitarias.

Uso en el proyecto: Este tipo de pruebas se han usado para comprobar todas las funcionalidades del sistema, pudiendo destacar las funcionalidades de las partidas donde por ejemplo se han simulado todas las peticiones que tendría que realizar un usuario para jugar una partida, por ejemplo se han hecho pruebas donde un usuario crea una partida en solitario y gana en un solo intento, en otras pruebas, el usuario gasta todos sus intentos y no acierta e incluso hay pruebas donde comete uno o dos fallos y posteriormente acierta.

- **Tipo de prueba:** Pruebas de interfaz.

Descripción: Las pruebas de interfaz se centran en evaluar la interacción del usuario con la aplicación a través de la interfaz. En estas pruebas nos aseguramos de que elementos visuales importantes se muestran correctamente y cumplen con los requisitos especificados.

Beneficios: Nos ayudan a detectar problemas de navegabilidad, errores visuales y problemas de usabilidad.

Uso en el proyecto: En el proyecto se han utilizado este tipo de pruebas en la mayoría de las funcionalidades, por ejemplo, se han utilizado en las partidas para asegurarnos de que se muestran los mensajes de fallo cuando se hace un intento y no se acierta el escudo, se han detectado fallos de navegación entre el registro y el inicio de sesión o se ha comprobado que, al actualizar el perfil de un jugador, los cambios se muestran inmediatamente.

- **Tipo de prueba:** Pruebas de integración.

Descripción: Este tipo de pruebas se usa para evaluar la interacción entre diferentes componentes o módulos del sistema.

Beneficios: Se busca encontrar problemas que puedan surgir cuando varios elementos software se combinan detectando así problemas en las comunicaciones entre módulos y procesos.

Uso en el proyecto: En el proyecto no se han desarrollado pruebas enfocadas directamente a comprobar la integración, pero sí que hay varias pruebas que lo hacen, un ejemplo de esto son las pruebas de interfaz del modo multijugador en las que se prueba la interacción entre diferentes módulos, por ejemplo para jugar una partida en línea primero se hace el registro de dos jugadores, después ambos inician sesión, uno crea una partida y el otro se une, posteriormente se juega la partida y finalmente se comprueba el resultado de la misma. En pruebas

como esta se puede comprobar la interacción entre el módulo de jugadores donde se encuentra el registro, el módulo de seguridad que es el que se encarga del inicio de sesión y el módulo de partidas en línea.

Además, para diseñar los casos de pruebas se han seguido distintas estrategias como pueden ser las siguientes:

- Cobertura CRUD, esta estrategia consiste en probar para cada entidad que persiste en la base de datos que se crea correctamente y se guarda en la base de datos, que una vez guardado se puede leer y actualizar correctamente y que tras todo esto se puede borrar. Esta estrategia se ha usado en cada entidad del proyecto, con la excepción de que algunas de las funcionalidades del proyecto no incluyen función de borrado ya que no era necesario por los requisitos, así que en esos casos solo se han probado las operaciones de creación, actualización y lectura.
- Errores conocidos, esta estrategia se basa en utilizar la experiencia para utilizar casos de uso que en otros proyectos han provocado fallos. Un ejemplo puede ser la experiencia de haber implementado funcionalidades como el inicio de sesión con certificado digital en otros proyectos, en esta funcionalidad se trabaja con un archivo multimedia y se detectaron fallos si se utilizaban archivos multimedia con un formato incorrecto o si se utilizaban archivos con el formato correcto, pero sin contenido. Esta experiencia ha sido utilizada para diseñar varios casos de prueba sobre la funcionalidad de añadir una nueva imagen.
- Valores límites, esta estrategia se basa en que los errores se suelen producir en los valores límite, por eso hay que hacer pruebas sobre dichos valores. Se puede ver la aplicación de esta estrategia en las pruebas que se hacen sobre la creación de nuevos usuarios, ya que se ha definido que las contraseñas deben tener una longitud de entre cuatro y treinta caracteres, por lo tanto, de esta restricción se pueden sacar los siguientes casos de prueba, tres casos positivos donde las contraseñas tienen cuatro, diez y treinta caracteres y dos casos negativos donde una tiene tres caracteres y la otra treinta y uno. Esta estrategia se puede aplicar a todo tipo de campo donde se hayan fijado restricciones.

Para automatizar la ejecución de las pruebas se ha creado un script de integración continua con la finalidad de que cada vez que se suba algún cambio a la rama principal del proyecto en Git Hub, se construya el proyecto y se ejecuten las pruebas automáticamente.

En el caso de que las pruebas se ejecuten satisfactoriamente se concluirá que no han surgido nuevos fallos y no ocurrirá nada, sin embargo, en el caso de que las pruebas fallen, las alertas saltaran por que se habrán detectado fallos y Git Hub lo notificará a través de un correo electrónico.

Además, en el apartado de Git Hub Actions se podrán ver los resultados de todas las veces que el script de integración continua se ha lanzado y en el caso de que alguna vez haya fallado, nos proporcionará información sobre la prueba que ha fallado y cuál ha sido el error.

All workflows				Filter workflow runs
				Event ▾ Status ▾ Branch ▾ Actor ▾
7 workflow runs				
✓ Refactor onlineGame	Java CI with Maven #7: Commit bc461de pushed by jucamgar2	master	5 days ago 1m 5s	...
✓ Refactor localGame and GameAlone	Java CI with Maven #6: Commit 6c1b82c pushed by jucamgar2	master	last week 1m 13s	...
✓ ChangePassword	Java CI with Maven #5: Commit 8b3d0f3 pushed by jucamgar2	master	last week 1m 12s	...
✓ Fix test assertions	Java CI with Maven #4: Commit dd1011b pushed by jucamgar2	master	last week 1m 5s	...
✓ Removed unused imports and xml scape in jsp	Java CI with Maven #3: Commit 4bb8c5d3 pushed by jucamgar2	master	last week 1m 6s	...
✓ Corrección Decimal format	Java CI with Maven #2: Commit 9d1cdff8 pushed by jucamgar2	master	last week 1m 13s	...
✗ Merge pull request #2 from jucamgar2/workflow	Java CI with Maven #1: Commit 21993db pushed by jucamgar2	master	last week 1m 1s	...

Figura 6.2: Ejemplo registro de Git Hub Actions

Finalmente, tras implementar todas estas pruebas y automatizando su ejecución se han detectado varios fallos que han sido corregidos, consiguiendo que las pruebas tengan una cobertura del código del 93%. Esto no significa que la aplicación esté exenta de fallos, pero sí que aporta confiabilidad al sistema ya que se han probado una gran cantidad de situaciones que pueden ocurrir en la aplicación y en todas estas situaciones la aplicación ha respondido tal y como se esperaba.

Capítulo 7 Comparación con otras alternativas

En la sección “Análisis de antecedentes y aportación realizada” se comentaron varias aplicaciones con una temática parecida a la de nuestra aplicación y se describió de una forma bastante simple algunas de las diferencias que tendría CUPES con respecto a las páginas ya existentes.

La principal característica que distingue a esta aplicación de las alternativas es el uso de imágenes como pistas. En la aplicación, solo se revelará una porción de nuevo de la pista en cada intento, a diferencia de la mayoría de las aplicaciones similares que muestran una imagen pixelada con variaciones en el grado de nitidez.

Esta alternativa nos lleva a la siguiente y es que para que la aplicación funcione correctamente se ha tenido que desarrollar un algoritmo de selección de pedazos para asegurarnos de que cada pedazo que se muestre contenga suficiente información como para que los usuarios puedan adivinar el escudo. Cuando decimos suficiente información, nos referimos a que al menos haya más de un color en la imagen y que el color dominante no supere el ochenta por ciento del pedazo. Sin embargo, en las páginas web comentadas anteriormente que utilizaban imágenes pixeladas como pistas, siempre la imagen se empezaba a volver más nítida desde el centro.

Además, por la forma en la que esta desarrollada la aplicación, dos partidas en las que tengamos que adivinar el mismo escudo, pueden llegar a ser completamente diferentes ya que cada escudo tiene dieciséis pedazos, de los cuales por lo general entre dos y cuatro son considerados no válidos, aunque hay escudos donde todos los pedazos son valido. Esto quiere decir que en una partida en la que un escudo tenga cuatro pedazos no válidos, en el peor de los casos, gastando todos nuestros intentos veremos un total de cinco pedazos del escudo, por lo que todavía nos quedarían por ver siete pedazos. Es completamente posible que en una futura partida nos vuelva a tocar adivinar el mismo escudo, si esto ocurre, todavía hay siete pedazos del escudo que nunca hemos visto individualmente, por lo que puede que no nos suenen.

Otra de las características que distingue a este proyecto de los demás es el hecho de que en la mayoría de las aplicaciones solo se puede jugar una sola partida al día, mientras que en CUPES no solo se podrán jugar a distintos modos de juego, sino que además se podrán jugar partidas ilimitadas en cada uno de ellos.

Al permitir a los usuarios jugar partidas ilimitadas a diario, podemos recopilar sus estadísticas y mostrarlas en un apartado de estadísticas con un ranking, lo que motivará

a nuestros jugadores a crearse una cuenta y jugar muchas partidas con ella para poder liderar el ranking.

Como ya hemos mencionado anteriormente, CUPES tiene dos modos de juego multijugador, característica que muy pocas aplicaciones tienen y por si esto fuera poco, en un modo dos jugadores pueden enfrentarse entre ellos desde el mismo dispositivo, mientras que en el otro modo dos jugadores cuales quiera en el mundo pueden enfrentarse en una partida en línea.

El hecho de tener un modo de juego multijugador requiere de un poco más de seguridad de la que tienen el resto de las aplicaciones ya que tenemos que asegurarnos de que una partida entre dos jugadores solo puede ser accesible para estos dos jugadores, por lo que nadie debe hacerse pasar por uno de ellos para manipular sus acciones en una partida. Esta misma seguridad ha sido implementada en todos los modos de juego, por lo que una partida en solitario solo será accesible para el usuario que la crea sin importar que este usuario no esté identificado y en las partidas uno contra uno local solo será accesible desde un solo dispositivo.

Por último, la aplicación tendrá un módulo de administración que permitirá a los usuarios con privilegios de administrador añadir más imágenes para las partidas, habilitar y deshabilitar a un usuario para que no pueda iniciar sesión. Además, el módulo de administración tiene estadísticas relativas a las imágenes que darán información a los administradores sobre los tipos de escudos que tienen más porcentaje de acierto, esto permitirá a los administradores balancear la dificultad de la aplicación ya que con esta información podrán tomar decisiones como habilitar o deshabilitar algunas imágenes.

Esta característica puede que esté disponible en muchas de las aplicaciones ya existentes, pero en ninguna es accesible y no se sabe que características tiene dicho modulo, sin embargo, en mi aplicación el módulo de administración será accesible si iniciamos sesión con las credenciales 'admin':'1111ñ@A?'.

Capítulo 8 Problemas encontrados

A lo largo del desarrollo de este proyecto se han ido encontrando diferentes dificultades por eso en este capítulo se tratarán de exponer todos los problemas que han ido surgiendo y las soluciones que se han tomado para resolverlos.

Antes de comenzar el trabajo, en el proceso de búsqueda de un Trabajo de Fin de Grado, me planteé la siguiente cuestión, ¿Debería buscar un proyecto donde tuviese que aprender una nueva tecnología? ¿O quizás sea mejor un proyecto donde pueda aumentar los conocimientos que ya tengo en alguna tecnología? Tras dedicar bastante tiempo a tomar esta decisión decidí afrontarlo desde un punto de vista profesional ya que el puesto en el que me gustaría trabajar tras terminar la carrera sería desarrollador Backend utilizando Spring Boot.

A pesar de que ya conocía esta tecnología, solo había trabajado con su versión 2.2.4 y cuando comencé el proyecto se encontraba en la versión 3.2 por lo que temía que mis conocimientos estuvieran muy desactualizados y por eso finalmente opté por desarrollar este proyecto con Spring Boot en su versión 3.2 para así actualizar mis conocimientos.

Una vez que comencé a trabajar en el proyecto fue cuando fui consciente de la cantidad de cambios que habían surgido en el framework, por eso mismo, la tarea 'Selección de tecnologías e inicialización del proyecto' fue estimada con veinte horas ya que en ella incluí el tiempo que necesitaría para aprender a utilizar esta nueva versión del framework. Para este aprendizaje utilice la documentación oficial proporcionada por Spring^{x xi xii xiii}.

Tras seleccionar la tecnología y contactar con mi tutor para realizar este proyecto llegó el momento de comenzar a trabajar en el mismo y justo entonces fue cuando surgió el siguiente problema y es que en un inicio pensé en escoger la temática que tendrían las imágenes que utilizaría la aplicación y por eso surgieron varias temáticas como pudieran ser clubes deportivos o lugares emblemáticos de Sevilla. Sin embargo, los lugares emblemáticos de Sevilla fue la primera temática que descarté ya que tras hacer varias pruebas comprobé que la dificultad del juego podría volverse excesiva, además de que automatizar que pedazos se muestran y comprobar que estos pedazos tengan suficiente información como para acertar se podía volver una tarea muy complicada y con mucho coste computacional a no ser que se realizase a mano y se escogieran manualmente los pedazos que se pueden mostrar. Finalmente, por las mismas razones también acabé destacando la temática de personas famosas. Por esta razón, la

aplicación finalmente acabó utilizando imágenes de clubes deportivos, concretamente equipos de fútbol para acotar el alcance de la aplicación.

Incluso durante una etapa bastante avanzada del desarrollo pensé en añadir alguna temática extra más allá de los escudos de fútbol como podrían ser escudos de otros deportes o cuadros de arte, pero decidí continuar con solo una temática para que la aplicación tuviese más coherencia.

Otro aspecto que ha presentado bastantes dificultades ha sido el protocolo de selección de pedazos ya que por la forma en la que está implementado revisa la porción de imagen seleccionada y comprueba si entre sus píxeles hay un color que predomina por encima de un 80% de los píxeles y si hay un color que ocupe más del 80% de los píxeles, se escoge un nuevo pedazo.

Una vez implementado el protocolo se desarrollaron pruebas y se comprobó que funcionaba correctamente en muchos casos, pero al repetir la prueba para distintas imágenes se detectaron ciertos casos en los que el algoritmo fallaba al determinar si un pedazo era válido o no.

Tras realizar muchas pruebas buscando la causa del fallo se detectó que el problema residía en las imágenes y es que en imágenes donde el fondo a simple vista es blanco, realmente si ampliamos la imagen el color blanco no es uniforme, por lo que la aunque un pedazo a simple vista se vea completamente blanco y del mismo color, realmente tiene 10 tipos de blanco distintos y ningún color representa el 80% de la imagen, por lo tanto el algoritmo en este tipo de imágenes no se comporta como se espera.

Para solucionar el problema simplemente se optó por utilizar todas las imágenes en formato png sin fondo y convertir la capa de transparencia en una capa de un color blanco uniforme. Tras cambiar las imágenes el algoritmo se empezó a comportar a la perfección en todos los casos.

Finalmente, también se han producido muchos problemas en el despliegue del proyecto, esto se ha debido a la dificultad para encontrar una plataforma donde desplegar el proyecto de forma automática. En una primera fase el proyecto se desplegó en la plataforma Zeabur <https://zeabur.com/>, pero los servidores donde se desplegó la aplicación estaban bastante lejos y eran lentos, por lo que la aplicación tenía problemas de carga, simplemente para cargar la interfaz inicial se podían necesitar tres segundos, así que tuve que buscar una nueva plataforma donde desplegar la aplicación. Tras un tiempo de búsqueda descubrí que Amazon Web Services <https://aws.amazon.com/>

tenía un plan gratuito y además en ese plan, la capacidad que ofrecían era bastante superior a las del resto de plataformas gratuitas, ya que, entre otras cosas, permiten escoger en qué servidores quieras desplegar tu aplicación, lo que ayuda a paliar los problemas de carga.

Tras desplegar la aplicación en Amazon Web Services los problemas de carga se solucionaron y ahora la aplicación está disponible en <http://cupes.es>. Otro problema que fue solucionado gracias al despliegue de la aplicación en Amazon Web Services fue el almacenamiento ya que en la primera plataforma la cantidad de almacenamiento era bastante limitada y aunque la aplicación cumplía con los límites de almacenamiento, si en un futuro se añadiesen más escudos se sobrepasarían estos límites y la aplicación dejaría de estar disponible, sin embargo, actualmente la aplicación tiene una capacidad de almacenamiento de treinta gigas, por lo que se podrían añadir diez mil escudos más a la aplicación y seguiría sobrandonos espacio.

Capítulo 9 Conclusiones y desarrollos futuros

Para concluir en este apartado haremos una recapitulación de las conclusiones obtenidas a lo largo del desarrollo de la aplicación web y tras el análisis de datos que se han realizado de la aplicación. Además, se propondrán futuras implementaciones que se podrían realizar para mejorar la implementación.

De cara al futuro uno de los aspectos de la aplicación que se podrían mejorar es el modo de juego en línea, y es que se podría cambiar la forma de iniciar una partida, y hacer el hecho de crear o unirse a una partida sea transparente para los usuarios, es decir que los usuarios simplemente pulsen sobre un botón con el texto “Jugar en línea” y que la aplicación sea quien se encargue de unir al jugador a una partida ya creada o en caso de no haber partidas creadas, crear una partida a nombre de este jugador para que otro pueda unirse.

Esto no ha sido implementado por una cuestión temporal ya que la idea de hacer esta mejora surgió en una etapa muy avanzada del proyecto donde todo ya estaba planificado e implementado y la realización de esta modificación hubiera provocado una desviación considerable en los costes. Además, este método para el emparejamiento tiene problemas ya que dos personas podrían ponerse de acuerdo y pulsar el botón jugar al mismo tiempo y en ese caso, muy probablemente el juego los hubiese emparejado para que jugasen entre ellos ya que perderíamos el factor de emparejamiento aleatorio que tenemos actualmente. Este problema podría ser aprovechado por algún usuario para crear dos cuentas y desde dos dispositivos distintos intentar jugar partidas donde solo participe él, asegurándose así de aumentar sus victorias.

En caso de desarrollar esta modificación en un futuro, habría que estudiar más alternativas para mejorar el proceso de emparejamientos y hacer que de alguna forma se vuelva aleatorio.

Otro desarrollo que se podría hacer en un futuro es añadir un modo donde aparezcan escudos retro. Para esta funcionalidad se podrá hacer uso de los datos estudiados para detectar cuales son los escudos más conocidos y a partir de estos datos se puede diseñar un modo de juego para un jugador donde para estos clubes más conocidos aparezcan sus escudos, pero del pasado. Por ejemplo, para el Real Madrid podrían aparecer los siguientes escudos:



Figura 9.1: Ejemplo de escudos retro

Además, para motivar a los jugadores a jugar más se podría hacer que modos de juego como este que se acaba de proponer o el modo de juego en línea estuvieran bloqueados hasta que se jugase cierto número de partidas en el modo de juego para un jugador. Esto impulsaría a los jugadores a crearse una cuenta y además jugar varias partidas para desbloquear nuevos modos de juego.

Esto no ha sido implementado ya que para el proyecto no tenía sentido denegar el acceso a las cuentas que no hubiesen jugado cierto número de partidas, ya que en la presentación del proyecto quienes creasen una cuenta no tendrían acceso a ciertos modos de juego hasta pasado un tiempo.

Tras comentar estos posibles desarrollos que se podrían implementar en un futuro, es momento de recapitular un poco y analizar el resultado final del proyecto.

En líneas generales se han cumplido los objetivos ya que se ha conseguido desarrollar una aplicación web donde se pueda jugar al juego Con Un Pedazo Es Suficiente, además dicha aplicación web se puede usar desde distintos tipos de dispositivos ya que la interfaz se adapta tanto a ordenadores como a móviles. Por otro lado, se ha reutilizado la lógica principal para crear dos modos de juego como lo son el modo de juego uno contra uno local y el modo de juego en línea.

Dentro de la aplicación hay un módulo con estadísticas que incitará a los jugadores a jugar más partidas y además también hay un módulo de administración que permitirá regular la aplicación desde la interfaz ya que este módulo permitirá añadir nuevas imágenes, deshabilitar o habilitar las existentes e incluso deshabilitar a los jugadores.

Por otro lado, me siento muy satisfecho con el proyecto ya que uno de mis objetivos personales era gestionar un proyecto desde cero y trabajar en todas las actividades que requiere el proceso de desarrollo software, como pueden ser la elicitation de requisitos, el diseño, el desarrollo, las pruebas o el despliegue, gracias a

este proyecto he podido experimentar todo el ciclo de desarrollo del software y me siento bastante contento con esto mismo.

Además, el proyecto me ha permitido formarme aún más en uno de mis frameworks de desarrollo Backend favoritos como lo es Spring boot, concretamente he podido actualizar mis conocimientos ya que yo me había formado en la versión 2.4 y actualmente ya está en la versión 3.2 que es la que he utilizado en este proyecto. Este aprendizaje me vendrá muy bien de cara a mi salida al mundo laboral.

Finalmente, otro de los objetivos de este proyecto era recopilar las estadísticas de la aplicación para poder procesarlas posteriormente y poder sacar conclusiones a partir de ellas, esto se ha podido lograr gracias a todas las personas que han utilizado la aplicación en este periodo y las conclusiones que se han podido sacar tras 304 partidas son las siguientes:

Ratio de acierto por país:

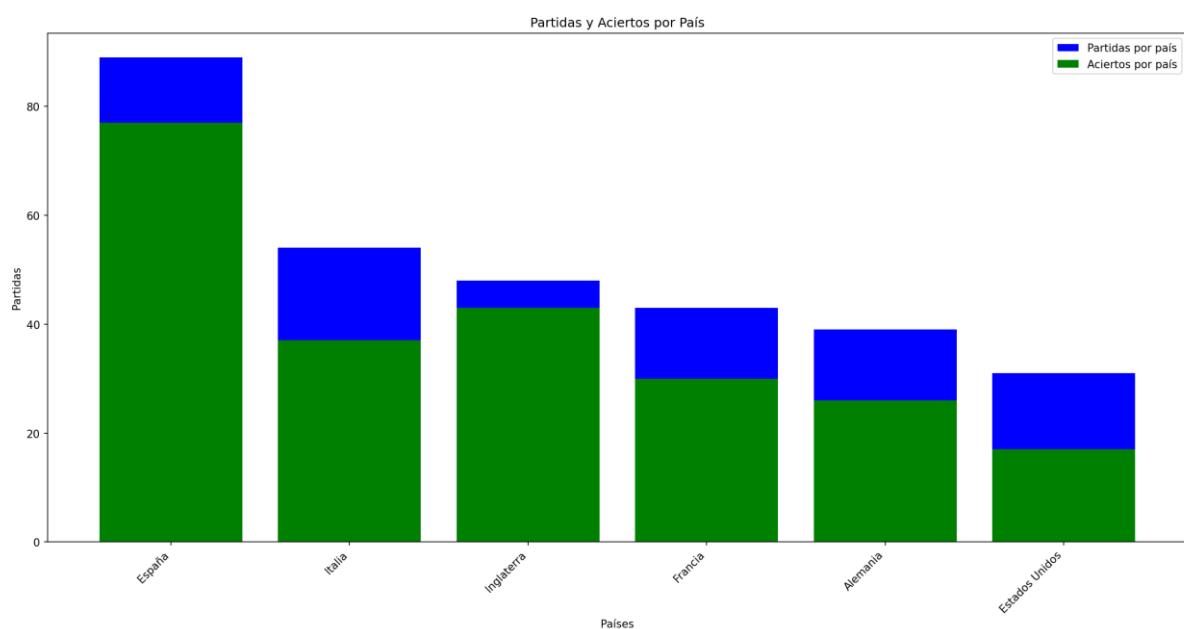


Figura 9.2: Gráfica de barras con los aciertos por país

País	Ratio
España	86%
Inglaterra	90%
Francia	70%
Italia	69%
Alemania	66%
Estados Unidos	54%

Cuadro 9.1: Ratio de acierto por país

Ratio de acierto por género

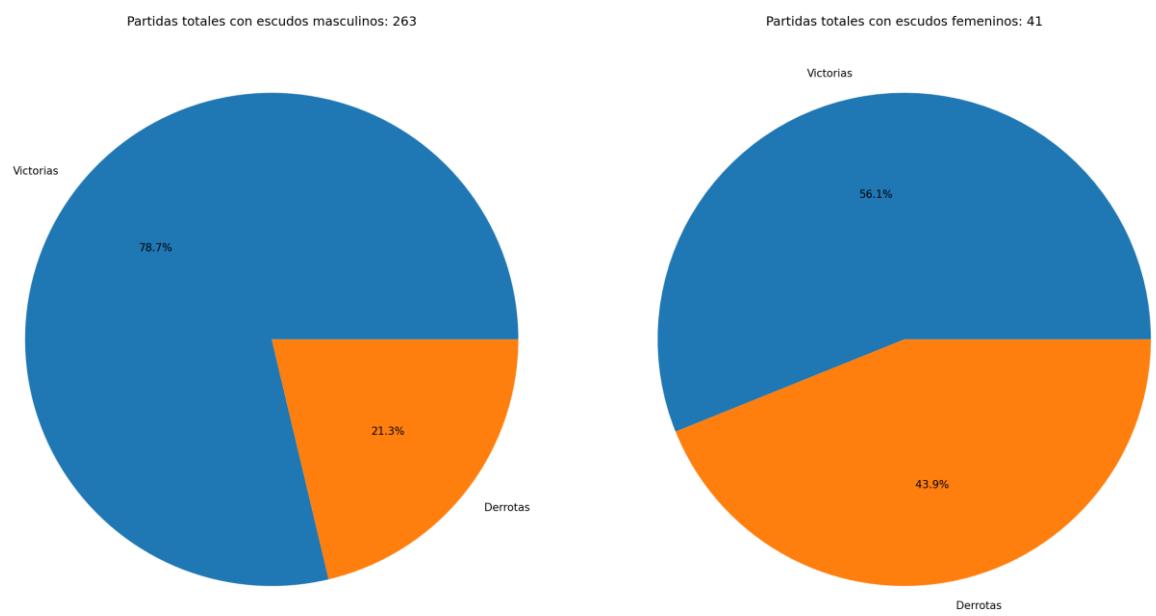


Figura 9.3: Ratio de acierto por género

Ratio de acierto si el escudo incluye el año:

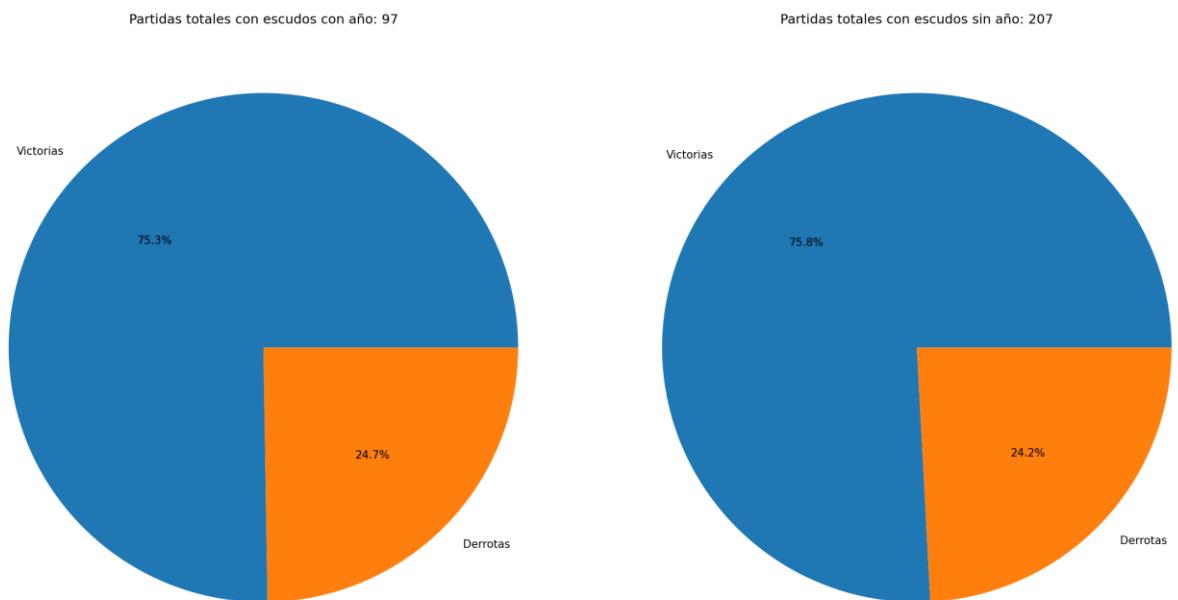


Figura 9.4: Ratio de acierto si el escudo incluye el año

Ratio de acierto si el escudo incluye las iniciales del club:

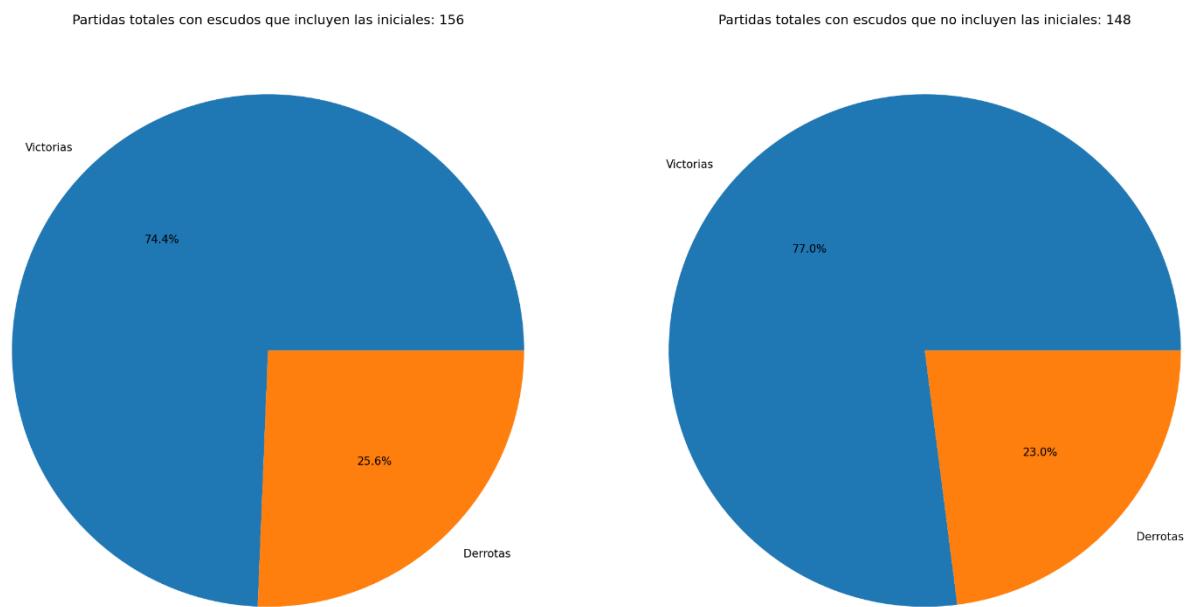


Figura 9.5: Ratio de acierto si el escudo incluye las iniciales del club

Ratio de acierto si el escudo incluye el nombre o parte de él:

Partidas totales con escudos que incluyen el nombre: 184

Partidas totales con escudos que no incluyen el nombre: 120

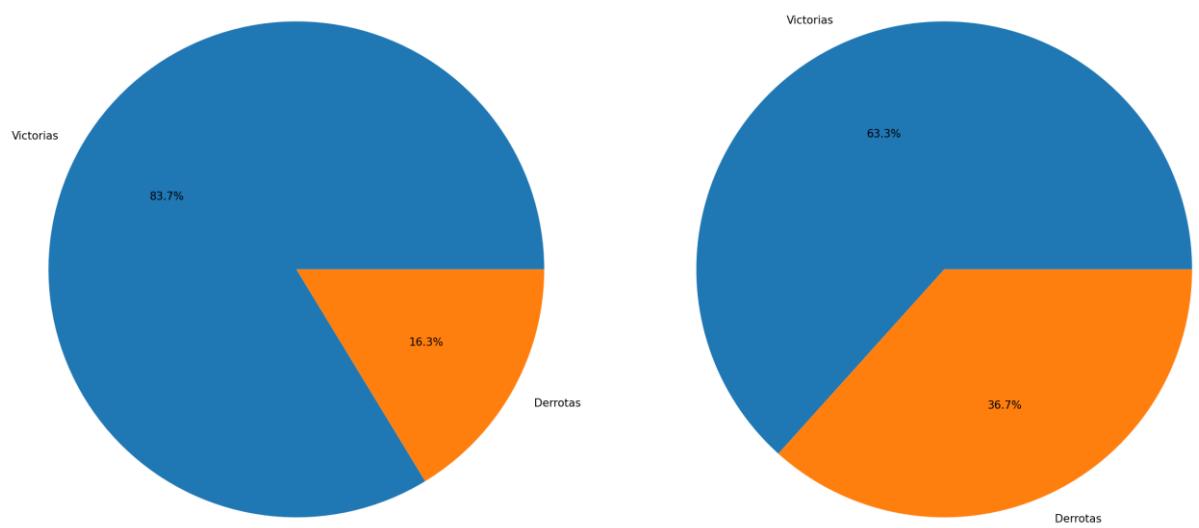


Figura 9.6: Ratio de acierto si el escudo incluye el nombre

Top 10 equipos que más han aparecido:

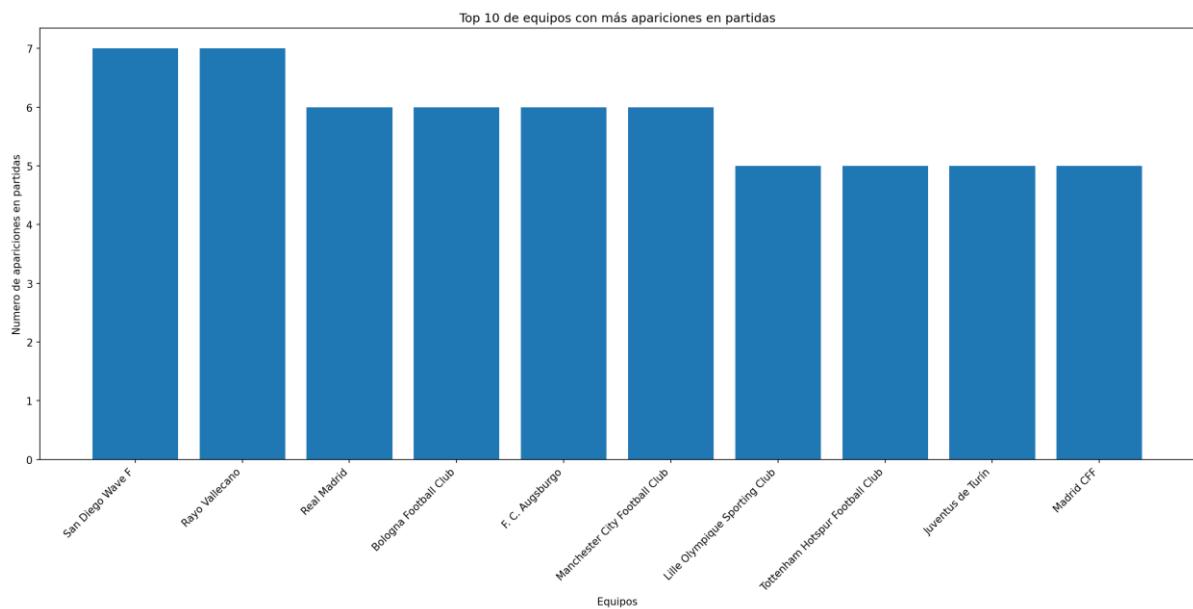


Figura 9.7: Top 10 equipos con más apariciones

Top 10 equipos que más veces han sido adivinados

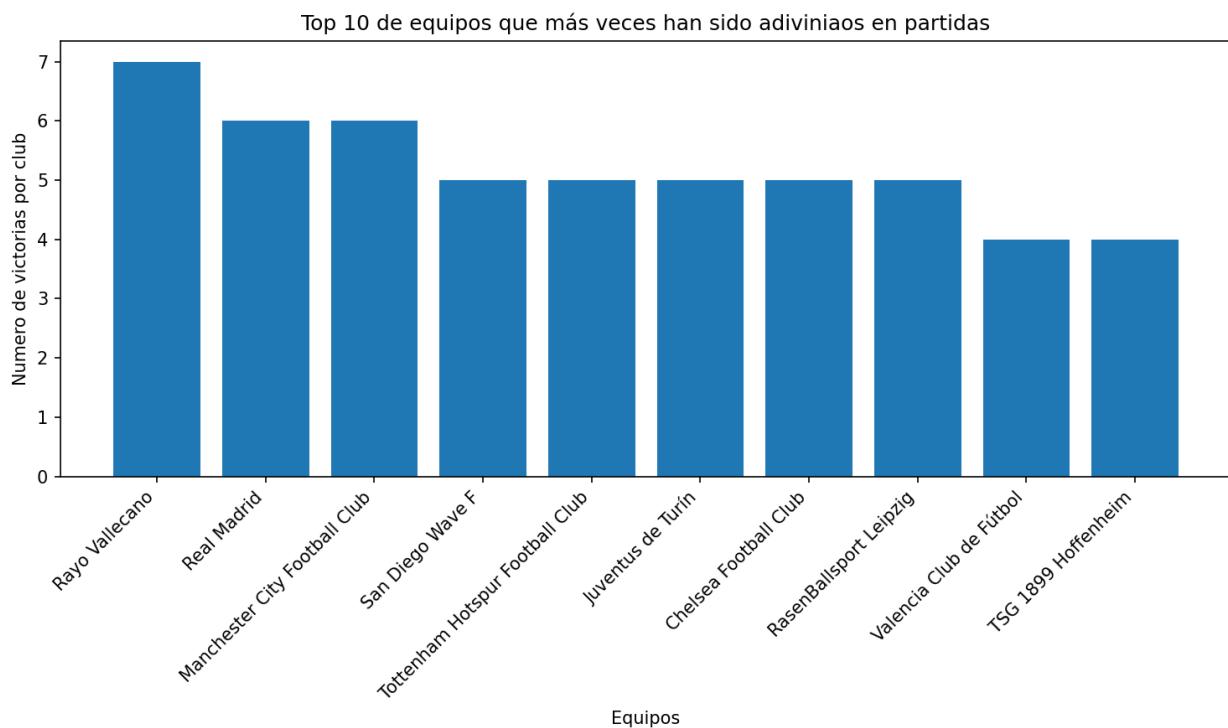


Figura 9.8: Top 10 equipo con mayor ratio de aciertos

Tras analizar todos estos datos podemos sacar las siguientes conclusiones:

- El país de donde los escudos han sido más conocidos es Inglaterra, seguido de España y los clubes menos reconocidos han sido los de Estados Unidos. Esto se debe principalmente a que la liga española y la inglesa son dos de las ligas más conocidas del mundo, mientras que de Estados Unidos solo se han incluido clubes de fútbol femenino ya que la liga femenina de Estados Unidos es una de las ligas femeninas de fútbol más conocidas, a pesar de esto el fútbol femenino hoy en día es menos visualizado a nivel mundial y esto se puede ver en el siguiente punto.
- También podemos concluir que los clubes de fútbol femenino son menos reconocidos que los de fútbol masculino, pero a pesar de esto, en más del cincuenta por ciento de las partidas se ha conseguido acertar el club de fútbol femenino lo que demuestra que poco a poco el fútbol femenino va ganando terreno y se va haciendo más conocido.
- En cuanto a las características de los escudos, podemos observar que el hecho de que un escudo incluya el año de fundación o las iniciales del club no tiene un efecto muy grande a la hora de que nuestros usuarios puedan adivinar el escudo, mientras que si el escudo contiene el nombre del equipo o parte de él, nuestros

usuarios tienen un 20% más de probabilidades de acertar respecto a escudos que no incluyan ninguna referencia al nombre del equipo.

- Finalmente, en cuanto a los clubes más reconocidos de la aplicación podemos destacar clubes que a nivel internacional son muy famosos como el Real Madrid, Manchester City, Chelsea o Juventus de Turín, junto a clubes cuyos escudos tienen formas muy reconocibles como el Valencia, Rayo Vallecano, Leipzig o Tottenham, acompañado por un club de fútbol femenino como lo es el San Diego Wave y un club cuyo escudo contiene el nombre completo como lo es el Hoffenheim.

En resumen, este Trabajo de Fin de Grado me ha servido para aplicar los conocimientos adquiridos en las distintas asignaturas cursadas a lo largo del grado y para profundizar en aún más en estos conocimientos, proporcionándome un valioso aprendizaje de cara a mi futuro profesional. Finalmente me gustaría destacar estos conocimientos que más he aplicado o profundizado a lo largo del desarrollo de este proyecto:

- Elicitación de requisitos.
- Diseño de aplicaciones web.
- Desarrollo de aplicaciones web.
- Desarrollo Backend con Spring Boot 3.2 y Java 17.
- Gestión de proyectos.
- Automatización de pruebas software.

Capítulo 10 Bibliografía

-
- ⁱ Colaboradores de Wikipedia. (2023, 14 noviembre). Modelo–Vista–Controlador. Wikipedia, la Enciclopedia Libre. <https://es.wikipedia.org/wiki/Modelo%20vista%20controlador>
- ⁱⁱ Last, First. (2024) Front Controller | Marco de Desarrollo de la Junta de Andalucía <https://www.juntadeandalucia.es/servicios/madeja/contenido/recurso/114>
- ⁱⁱⁱ Sending Sql Statements. (2024) The Repository Design Pattern <https://www.umlboard.com/design-patterns/repository.html>
- ^{iv} Wikipedia contributors. (2023, 2 marzo). Service layer pattern. Wikipedia. https://en.wikipedia.org/wiki/Service_layer_pattern
- ^v Wikipedia contributors. (2024, 10 febrero). Dependency injection. Wikipedia. https://en.wikipedia.org/wiki/Dependency_injection
- ^{vi} Colaboradores de Wikipedia. (2023, septiembre 27). *Proxy (patrón de diseño)*. Wikipedia, la Enciclopedia Libre. [https://es.wikipedia.org/wiki/Proxy_\(patr%C3%B3n_de_dise%C3%BAo\)](https://es.wikipedia.org/wiki/Proxy_(patr%C3%B3n_de_dise%C3%BAo))
- ^{vii} Iluwatar. (s. f.-b). *Domain model*. Java Design Patterns. <https://java-design-patterns.com/patterns/domain-model/>
- ^{viii} The Primary Key. (2024) Source Code Examples. <https://www.sourcecodeexamples.net/2018/05/identity-field-pattern.html>
- ^{ix} Toxboe, A. (s. f.). Pagination design pattern. Copyright (C) 2007-2024 Anders Toxboe. <https://ui-patterns.com/patterns/Pagination>
- ^x Phillip Webb, D.S. (no date) Spring Boot Reference Documentation <https://docs.spring.io/spring-boot/docs/current/reference/htmlsingle/>
- ^{xi} New features and enhancements in Spring Framework 3.2. <https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/new-in-3.2.html>
- ^{xii} Appendix D. migrating to Spring Framework 3.2. <https://docs.spring.io/spring-framework/docs/3.2.x/spring-framework-reference/html/migration-3.2.html>

^{xiii} Getting started: Securing a web application (no date) Getting Started |Securing a Web Application. <https://spring.io/guides/gs/securing-web>